

**ANKARA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**DOKTORA TEZİ**

**MOBİL UÇ BİLİŞİM'DE HESAPLAMA AKTARIM SORUNU İÇİN  
YENİ BİR YAKLAŞIM**

**Hasan Faruk ÇOBAN**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**ANKARA  
2026**

**Her hakkı saklıdır**

## ÖZET

Doktora Tezi

### MOBİL UÇ BİLİŞİM'DE HESAPLAMA AKTARIM SORUNU İÇİN YENİ BİR YAKLAŞIM

Hasan Faruk ÇOBAN

Ankara Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Mehmet Serdar GÜZEL

Bulut bilişim, kısa sürede bilişim teknolojileri alanında önemli bir dönüşüm yaratarak yüksek işlem gücü gerektiren uygulamaların daha geniş kullanıcı kitleleri tarafından erişilebilir hale gelmesini sağlamıştır. Ölçeklenebilir ve esnek kaynak yapısı sayesinde, sınırlı donanım imkanlarına sahip kullanıcılar dahi karmaşık hesaplama işlemlerini gerçekleştirebilmektedir. Bu gelişme, gerçek zamanlı sensör verisi işleme, anlık görüntü analizi ve büyük veri uygulamaları gibi yoğun hesaplama gerektiren sistemlerin yaygınlaşmasına önemli katkı sağlamıştır.

Fakat bu yeni durum kendi içinde yeni problemleri de getirmiştir. Hesaplama ve işlemlerin buluta aktarılması ve geri dönmesi belli bir zaman almaktadır. Bilgisayar ağlarının önemli problemlerinden olan gecikme (delay) burada da karşımıza çıkmıştır. Bu problemi çözmek için mobil uç bilişim ortaya atılmıştır. Özellikle 5G teknolojileri ile uyumlu çalışması planlanan bu mimaride uzaktaki merkezi bulut bilişim sunucularına hesaplama aktarımı yerine kullanıcılara daha yakın mobil uç sunucular konumlandırılarak arada geçen gecikme süresinin kısaltılması amaçlanmaktadır. Bir nevi vekil sunucu görevi gören mobil uç bilişimde iki önemli süreç vardır. Bunlardan ilki sunucuların optimum konumlandırılması ikincisi hesaplama aktarımıdır. Bu tez, hesaplama aktarım süreci için en uygun yöntemin geliştirilmesini amaçlamaktadır. Bu kapsamda hem veri üretimini hem de sistem altyapısını destekleyen yenilikçi bir deney düzeneği ve yazılım altyapısı tasarlanarak gerçekleştirilmiştir. Geliştirilen bu altyapı kullanılarak farklı hesaplama aktarım yöntemleri uygulanmış, analiz edilmiş ve sonuçları karşılaştırmalı olarak sunulmuştur.

**Haziran 2026, 158 sayfa**

**Anahtar Kelimeler:** Bulut Bilişim, Mobil Uç Bilişim, Nesnelerin İnterneti, Sezgisel Algoritmalar

## ABSTRACT

PhD Thesis

### A NEW APPROACH FOR COMPUTATIONAL OFFLOADING PROBLEM IN MOBILE EDGE COMPUTING,

Hasan Faruk ÇOBAN

Ankara University  
Graduate School of Natural and Applied Sciences  
Department of Computer Engineering

Supervisor: Prof. Dr. Mehmet Serdar GÜZEL

Cloud computing has had a significant impact on the computing world in a very short time. By enabling complex and large-scale operations to be performed outside of supercomputers, it has provided solutions to the needs of users with limited resources. This capability has made certain applications, such as real-time sensor data processing and real-time image processing, possible, which were previously considered unattainable. However, this new scenario has brought its own set of challenges. The transfer of computations and processes to the cloud and back requires a certain amount of time. Latency, one of the critical issues in computer networks, has also emerged in this context. To address this problem, multi-access edge computing (MEC) has been introduced. This architecture, designed to work in harmony with 5G technologies, aims to reduce latency by positioning mobile edge servers closer to users instead of transferring computations to distant centralized cloud servers. In mobile edge computing, which acts as a sort of proxy server, two key processes are involved. The first is the optimal placement of servers, and the second is computation offloading. This thesis aims to develop an optimal method for the computation offloading process. In this context, simulation setup for both data creation and infrastructure is designed. Via simulation optimal methods are analyzed.

**June 2026, 158 pages**

**Key Words:** Cloud Computing, Multi-Access Edge Computing, Internet of Things, Heuristic Algorithms

## TEŞEKKÜR

Tez çalışmam boyunca, bilgisiyle sürekli destekleyen, sabrını esirgemeyen ve beni cesaretlendiren danışman Hocam Sayın Prof. Dr. Mehmet Serdar GÜZEL'e,

Tez izleme komitemin üyeleri olan, değerli yorum ve katkıları ile çalışmanın niteliğini arttıran ve sabırla beni yönlendiren sayın Prof. Dr. Asım Egemen YILMAZ ve Doç. Dr. Özgür TANRIÖVER'e

Maddi manevi desteklerini esirgemeyen, her daim yanımda olan anneme, babama, bana sürekli daha iyisini yapma motivasyonu veren oğlum Yiğit'e en derin duygularla teşekkür ederim.

Hasan Faruk ÇOBAN  
Ankara, Haziran 2026

## İÇİNDEKİLER

### TEZ ONAY SAYFASI

|                                                                               |     |
|-------------------------------------------------------------------------------|-----|
| ETİK .....                                                                    | i   |
| ÖZET .....                                                                    | ii  |
| ABSTRACT .....                                                                | iii |
| TEŞEKKÜR.....                                                                 | iv  |
| ŞEKİLLER DİZİNİ.....                                                          | x   |
| ÇİZELGELER DİZİNİ .....                                                       | xii |
| 1. GİRİŞ .....                                                                | 1   |
| 1.1 Genel Bilgi.....                                                          | 1   |
| 1.1.1 Tez kapsamının sınırları ve odak noktası .....                          | 2   |
| 1.1.2 Okuyucu için kavram haritası .....                                      | 3   |
| 1.2 Motivasyon ve Tez Kapsamı .....                                           | 3   |
| 1.2.1 Meta-sezgisel yaklaşımın gerekçesi .....                                | 6   |
| 1.2.2 MEC bağlamında meta-sezgisel yaklaşım .....                             | 6   |
| 1.3 Tezin Yapısı .....                                                        | 7   |
| 1.3.1 Araştırma soruları .....                                                | 8   |
| 1.3.2 Araştırma hedefleri.....                                                | 9   |
| 1.3.3 Bilimsel katkılar .....                                                 | 9   |
| 1.3.4 Tez bölümlerine genel bakış.....                                        | 10  |
| 1.3.5 Deneysel kapsam ve metodolojik bütünlük .....                           | 11  |
| 2. KAYNAK ÖZETLERİ VE KURAMSAL TEMELLER .....                                 | 12  |
| 2.1 Nesnelerin İnterneti ve Bulut Bilişim .....                               | 12  |
| 2.2 Hesaplama Aktarım.....                                                    | 14  |
| 2.2.1 IoT–Bulut–MEC zinciri ve hesaplama aktarımına geçiş .....               | 15  |
| 2.2.2 Hesaplama aktarımında sezgisel yöntemlerin konumu.....                  | 16  |
| 2.2.3 Aktarım kararında meta-sezgisellerin rolü ve RL ile konumlandırma ..... | 17  |
| 2.2.4 Yerel, tam ve kısmi aktarımın bağlamdaki yeri .....                     | 18  |
| 2.3 Kaynak Tahsisi.....                                                       | 19  |
| 2.3.1 Kaynak tahsisi ile hesaplama aktarımı arasındaki ilişki.....            | 21  |
| 2.4 Sunucu Konumlandırma .....                                                | 22  |
| 2.4.1 Sunucu konumlandırmanın aktarım kararına dolaylı etkisi.....            | 23  |
| 2.5 Ek Yük Yönetimi.....                                                      | 24  |

|                                                                                             |    |
|---------------------------------------------------------------------------------------------|----|
| 2.5.1 Ek yükün $F(x)$ modeline yansımaları .....                                            | 25 |
| 2.6 Literatür Taraması Özeti .....                                                          | 26 |
| 2.6.1 Dört zorluk tablosunun sentezi ve tezdeki konumu .....                                | 28 |
| 2.6.2 Gerçek dünya veri setleri ve MEC literatürü .....                                     | 29 |
| 2.6.3 EdgeCloudSim, iFogSim ve özel simülatör tercihi .....                                 | 30 |
| 2.6.4 Literatürden yöntemle geçiş: Boşluk ve tez konumlandırması .....                      | 31 |
| 3. MATERYAL VE YÖNTEM.....                                                                  | 33 |
| 3.1 Deneysel Araştırma Tasarımı, Geçerlilik ve Tutarlılık.....                              | 33 |
| 3.2 Veri Oluşturucu.....                                                                    | 33 |
| 3.2.1 Poisson( $\lambda$ ) dağılımı: Katsayılar, gerekçe ve on beş senaryo profili.....     | 40 |
| 3.2.2 Senaryo üreticinin meta-sezgisel algoritmalarla uygunluğu.....                        | 41 |
| 3.2.3 Değişken grupları ve aktarım kararına etkisi.....                                     | 43 |
| 3.3 Simülasyon Düzenegi .....                                                               | 43 |
| 3.3.1 Simülasyon bileşenlerinin işlevsel ayrıntıları.....                                   | 44 |
| 3.3.2 Bileşenlerin sorumlulukları .....                                                     | 45 |
| 3.4 Karar Algoritmaları .....                                                               | 46 |
| 3.4.1 Beş algoritma seçiminin gerekçesi ve ortak protokol.....                              | 46 |
| 3.4.2 Genetik algoritma (Genetic Algorithm) .....                                           | 47 |
| 3.4.3 Genetik algoritma (GA) — MEC aktarım probleminde uygulama adımları.....               | 48 |
| 3.4.4 Parçacık sürüsü optimizasyonu (Particle Swarm Optimization).....                      | 49 |
| 3.4.5 Parçacık sürüsü optimizasyonu (PSO) — MEC aktarım probleminde uygulama adımları.....  | 50 |
| 3.4.6 Gri kurt algoritması (Gray Wolf Algorithm).....                                       | 52 |
| 3.4.7 Gri kurt optimizasyonu (GWO) — MEC aktarım probleminde uygulama adımları.....         | 53 |
| 3.4.8 Karınca kolonisi optimizasyonu (Ant Colony Optimization (ACO)).....                   | 54 |
| 3.4.9 Karınca kolonisi optimizasyonu (ACO) — MEC aktarım probleminde uygulama adımları..... | 56 |
| 3.4.10 Yapay arı kolonisi (Artificial Bee Colony).....                                      | 57 |
| 3.4.11 Yapay arı kolonisi (ABC) — MEC aktarım probleminde uygulama adımları.....            | 57 |
| 3.4.12 Meta-sezgisel algoritmaların MEC aktarım problemine uygulanması .....                | 59 |
| 3.4.13 MEC operatörü için meta-sezgisel karar döngüsü .....                                 | 60 |
| 3.4.14 Adil kıyas: arama limiti ve fitness değerlendirme maliyeti.....                      | 61 |

|                                                                                                |     |
|------------------------------------------------------------------------------------------------|-----|
| 3.4.15 Yakınsama, global optimum ve stokastik sınırlar.....                                    | 62  |
| 3.5 Amaç (Objective) Fonksiyonu.....                                                           | 62  |
| 3.5.1 Çok kriterli amaç fonksiyonunun formülasyonu ve ağırlıklandırma.....                     | 65  |
| 3.5.2 Gecikme ve kaynak metriklerinin simülatör çıktısı olarak türetilmesi.....                | 65  |
| 3.5.3 Amaç fonksiyonlarının analitik değerlendirilmesi .....                                   | 67  |
| 3.5.4 Karar uzayı, kısıtlar ve problem karmaşıklığı.....                                       | 67  |
| 3.6 Test Senaryoları ve Başarı Metrikleri.....                                                 | 68  |
| 3.6.1 Senaryo sınıflandırması: Statik/Dinamik ve kaynak profili.....                           | 68  |
| 3.6.2 Deney protokolü özeti ve metrik okuma.....                                               | 73  |
| 3.6.3 Gerçek dünya veri setleri ve veri seti deney modülleri .....                             | 73  |
| 3.6.4 Otuz iki değişken ile gerçek dünya veri setlerinin uyumluluğu .....                      | 75  |
| 3.6.5 Literatürde veri seti kullanımı ve bağlam uyumu .....                                    | 77  |
| 3.6.6 Üç deney modunun ayrıntılı metodolojisi.....                                             | 78  |
| 3.6.7 Metodolojik sınırlamalar ve gelecek iyileştirmeler .....                                 | 80  |
| 3.6.8 Tekrarlanabilirlik, ortam ve yazılım sürümleri .....                                     | 83  |
| 3.6.9 Hesaplama maliyeti ve deneysel ortam.....                                                | 83  |
| 4. BULGULAR VE TARTIŞMA .....                                                                  | 85  |
| 4.1 Test Sonuçları .....                                                                       | 85  |
| 4.1.1 Deneysel tasarım ve sonuç okuma ilkeleri.....                                            | 85  |
| 4.1.2 Senaryo sonuçlarının yorumlanması .....                                                  | 86  |
| 4.1.3 Gerçek dünya veri setleri ile deney sonuçları .....                                      | 103 |
| 4.1.4 Gerçek veri seti sonuçlarını okuma ilkeleri.....                                         | 104 |
| 4.1.5 Deneysel protokolün tekrarlanabilir tanımı .....                                         | 106 |
| 4.1.6 Veri seti başına ayrıntılı bulgu ve literatür yorumu .....                               | 108 |
| 4.1.7 Birleşik modlar: hibrit ve adaptör+kalibrasyon.....                                      | 110 |
| 4.1.8 Tekrarlı koşu protokolü ve sonuçları .....                                               | 111 |
| 4.1.9 İstatistiksel raporlama ve wilcoxon testi.....                                           | 112 |
| 4.1.10 Dataset-only çoklu test sonuçları .....                                                 | 113 |
| 4.1.11 Birleşik mod çoklu test sonuçları.....                                                  | 115 |
| 4.1.12 Poisson senaryoları ile gerçek dünya veri seti sonuçlarının çapraz karşılaştırması..... | 116 |
| 4.1.13 Çapraz bulgular ve AS1–AS4 bağlantısı.....                                              | 117 |
| 4.1.14 Literatür uyum değerlendirmesi (Veri Seti Kökenli Yığın).....                           | 118 |
| 4.1.15 Bulguların sentezi ve algoritma seçim çerçevesi .....                                   | 119 |

|                                                                          |            |
|--------------------------------------------------------------------------|------------|
| <b>4.2 Tartışma</b>                                                      | <b>120</b> |
| 4.2.1 Araştırma sorularına yanıtların özeti.....                         | 120        |
| 4.2.2 Genel sonuç ve seçim çerçevesi önerisi .....                       | 120        |
| 4.2.3 MEC'de meta-sezgisel kullanımın sağladığı faydaların sentezi ..... | 121        |
| 4.2.4 Poisson senaryo sonuçlarının değerlendirilmesi .....               | 122        |
| 4.2.5 Gerçek Dünya veri seti sonuçlarının değerlendirilmesi .....        | 126        |
| 4.2.6 Deneysel eksenlerin birlikte değerlendirilmesi .....               | 128        |
| 4.2.7 Önceki yayın bulguları ile karşılaştırma .....                     | 129        |
| 4.2.8 Çalışmanın sınırlamaları ve gelecek çalışmalar.....                | 130        |
| <b>5. SONUÇ</b> .....                                                    | <b>132</b> |
| <b>EKLER</b> .....                                                       | <b>143</b> |
| EK 1 Gerçek Dünya Veri Setleri, Atıflar ve Deney Protokolü .....         | 144        |
| EK 2 On Beş Senaryo ScenarioConfig Parametreleri .....                   | 145        |
| EK 3 Gerçek Dünya Veri Setleri Değişken Uyumluluğu.....                  | 151        |
| <b>ÖZGEÇMİŞ</b> .....                                                    | <b>158</b> |

## KISALTMALAR DİZİNİ

|       |                                                       |
|-------|-------------------------------------------------------|
| MEC   | Mobil Uç Bilişim (Multi-Access Edge Computing)        |
| IoT   | Nesnelerin İnterneti (Internet of Things)             |
| ML    | Makine öğrenimi (Machine Learning)                    |
| AI    | Yapay Zeka (Artificial Intelligence)                  |
| RL    | Takviyeli Öğrenme (Reinforcement Learning)            |
| DRL   | Derin Takviyeli Öğrenme (Deep Reinforcement Learning) |
| GSM   | Global System for Mobile Communications               |
| HTTP  | Hyper Text Transfer Protocol                          |
| GA    | Genetik Algoritma (Genetic Algorithm)                 |
| PSO   | Particle Swarm Optimization                           |
| GWO   | Gri Kurt Algoritması (Gray Wolf Algorithm)            |
| ABC   | Yapay Arı Kolonisi (Artificial Bee Colony)            |
| ACO   | Ant Colony Optimization                               |
| Wi-Fi | Wireless Fidelity                                     |

## ŞEKİLLER DİZİNİ

|                                                                                          |    |
|------------------------------------------------------------------------------------------|----|
| Şekil 1.1 Problem tanımı.....                                                            | 4  |
| Şekil 2.1 Hesaplama aktarım.....                                                         | 18 |
| Şekil 2.2 Kaynak kullanımı.....                                                          | 21 |
| Şekil 2.3 Sunucu konumlandırma .....                                                     | 23 |
| Şekil 2.4 Ek yük yönetimi.....                                                           | 25 |
| Şekil 3.1 Simülasyon mimarisi .....                                                      | 46 |
| Şekil 4.1 Baz senaryo için gecikme .....                                                 | 87 |
| Şekil 4.2 Baz senaryo için kaynak kullanımı .....                                        | 87 |
| Şekil 4.3 Yüksek yoğunluklu kentsel alan için gecikme.....                               | 88 |
| Şekil 4.4 Yüksek yoğunluklu kentsel alan için kaynak kullanımı .....                     | 88 |
| Şekil 4.5 Akıllı şehir ağı için gecikme .....                                            | 89 |
| Şekil 4.6 Akıllı şehir ağı için kaynak kullanımı.....                                    | 89 |
| Şekil 4.7 Tüketici ağı kullanımı sınırlı iş bölgesi için gecikme .....                   | 90 |
| Şekil 4.8 Tüketici ağı kullanımı sınırlı iş bölgesi için kaynak kullanımı .....          | 90 |
| Şekil 4.9 Etkinlik kaynaklı yüksek trafik için gecikme .....                             | 91 |
| Şekil 4.10 Etkinlik kaynaklı yüksek trafik için kaynak kullanımı .....                   | 91 |
| Şekil 4.11 Sınırlı internet erişimi olan kırsal bölge için gecikme .....                 | 92 |
| Şekil 4.12 Sınırlı internet erişimi olan kırsal bölge için kaynak kullanımı.....         | 92 |
| Şekil 4.13 Tarımsal ağ ihtiyaçları için kaynak kullanımı .....                           | 93 |
| Şekil 4.14 Tarımsal ağ ihtiyaçları için kaynak kullanımı .....                           | 93 |
| Şekil 4.15 Fiber optik veya 5G altyapısına sahip kırsal bölge için gecikme .....         | 94 |
| Şekil 4.16 Fiber optik veya 5G altyapısına sahip kırsal bölge için kaynak kullanımı..... | 94 |
| Şekil 4.17 Banliyö ofis ve konut ağı için gecikme.....                                   | 95 |
| Şekil 4.18 Banliyö ofis ve konut ağı için kaynak kullanımı .....                         | 95 |
| Şekil 4.19 Evden çalışma ve aile ağı için gecikme .....                                  | 96 |
| Şekil 4.20 Evden çalışma ve aile ağı için kaynak kullanımı.....                          | 96 |
| Şekil 4.21 Büyükşehirde veri merkezi performansı için gecikme .....                      | 97 |
| Şekil 4.22 Büyükşehirde veri merkezi performansı için kaynak kullanımı .....             | 97 |
| Şekil 4.23 Felaket kurtarma testi için gecikme .....                                     | 98 |
| Şekil 4.24 Felaket kurtarma testi için kaynak kullanımı.....                             | 98 |
| Şekil 4.25 Uzak bölgelerde uç bilişim ağı için gecikme.....                              | 99 |

|                                                                                        |     |
|----------------------------------------------------------------------------------------|-----|
| Şekil 4.26 Uzak bölgelerde uç bilişim ağı için kaynak kullanımı.....                   | 99  |
| Şekil 4.27 Kırsal veya uzak alanlarda geçici ağ kurulumları için gecikme .....         | 100 |
| Şekil 4.28 Kırsal veya uzak alanlarda geçici ağ kurulumları için kaynak kullanımı..... | 100 |
| Şekil 4.29 Kırsal alanlarda teletıp ve e-öğrenme için gecikme.....                     | 101 |
| Şekil 4.30 Kırsal alanlarda teletıp ve e-öğrenme için kaynak kullanımı .....           | 101 |
| Şekil 4.31 Ortalama gecikme .....                                                      | 103 |
| Şekil 4.32 Ortalama kaynak kullanımı.....                                              | 103 |
| Şekil 4.33 Dinamik senaryolar için gecikme .....                                       | 123 |
| Şekil 4.34 Dinamik senaryolar için kaynak kullanımı .....                              | 123 |
| Şekil 4.35 Statik senaryolar için gecikme .....                                        | 124 |
| Şekil 4.36 Statik senaryolar için kaynak kullanımı.....                                | 124 |

## ÇİZELGELER DİZİNİ

|                                                                               |     |
|-------------------------------------------------------------------------------|-----|
| Çizelge 2.1 Hesaplama aktarım makaleleri.....                                 | 27  |
| Çizelge 2.2 Kaynak tahsisi makaleleri .....                                   | 27  |
| Çizelge 2.3 Sunucu konumlandırma makaleleri .....                             | 28  |
| Çizelge 2.4 Ek yük yönetimi makaleleri .....                                  | 28  |
| Çizelge 2.5 Gerçek dünya veri setleri ve tezdeki kullanım amacı .....         | 30  |
| Çizelge 3.1 Coğrafi değişkenler .....                                         | 35  |
| Çizelge 3.2 Görev değişkenleri .....                                          | 36  |
| Çizelge 3.3 Yerel kaynak değişkenleri.....                                    | 37  |
| Çizelge 3.4 Uç sunucu değişkenleri .....                                      | 38  |
| Çizelge 3.5 Ağ değişkenleri .....                                             | 39  |
| Çizelge 3.6 Dış çevre değişkenleri.....                                       | 39  |
| Çizelge 3.7 Senaryo adları ve temsili $\lambda$ merkezleri.....               | 40  |
| Çizelge 3.8 Meta-sezgisel algoritma hiperparametreleri .....                  | 60  |
| Çizelge 3.9 Amaç fonksiyonu semboller ve tanımları .....                      | 64  |
| Çizelge 3.10 Rao (2009) ilkeleri ile $F(x)$ tanımının uyumu.....              | 65  |
| Çizelge 3.11 Veri seti kategorisi $\times$ değişken kategorisi (özet) .....   | 76  |
| Çizelge 3.12 Deney modları karşılaştırması .....                              | 79  |
| Çizelge 4.1 Harici deney protokolü .....                                      | 106 |
| Çizelge 4.2 Dataset-only: en iyi bireyin yanıt süresi $t_r$ (ms).....         | 107 |
| Çizelge 4.3 Dataset-only: kazanan yerleşim (LOCAL / EDGE) .....               | 107 |
| Çizelge 4.4 Dataset-only: gecikme ve yürütme bileşenleri (ms) .....           | 108 |
| Çizelge 4.5 Birleşik modlar: $t_r$ (ms) ve yerleşim .....                     | 110 |
| Çizelge 4.6 Veri seti kökenli tekrarlı koşu parametreleri .....               | 111 |
| Çizelge 4.7 Dataset-only: $t_r$ ortalama $\pm$ std (ms) .....                 | 113 |
| Çizelge 4.8 Dataset-only modda eşleştirilmiş Wilcoxon testi özeti (n=10)..... | 114 |
| Çizelge 4.9 Birleşik modlar: $t_r$ ortalama $\pm$ std (ms).....               | 115 |
| Çizelge 4.10 Erken makale koşusu: temsili gecikme ve kaynak sonuçları.....    | 130 |

# 1. GİRİŞ

## 1.1 Genel Bilgi

Mobil Uç Bilişim (Multi-Access Edge Computing–MEC), Bulut Bilişimdeki gecikme problemine çözüm olarak geliştirilmiş bir mimaridir. Bu mimaride, yüksek işlem gücüne sahip merkezler uçlara doğru yaygınlaştırılarak vekil sunucular aracılığıyla son kullanıcıya yaklaştırılması hedeflenmiştir. Nesnelerin İnterneti (Internet of Things, IoT) terimi ilk olarak 1999 yılında ortaya atılmıştır (Ashton 2009). 2010'ların başlarında önemli bir araştırma alanı haline gelen IoT, bilgisayar ağlarına entegre edilen çok sayıda cihazın ortaya çıkmasına neden olmuştur. Bu cihazların kaynak açısından sınırlı olması sebebiyle, Bulut Bilişim fikri geliştirilmiş ve kısa sürede yaygınlaşmıştır. Bulut Bilişimde, bir cihaz, kendi başına yapamayacağı ya da uzun zaman alacak işlemleri, ağa bağlı diğer cihazlara veya veri merkezlerine aktararak gerçekleştirebilir hale gelmiştir. Bu süreçte, veri merkezleri genellikle hizmet sağlayıcılar tarafından yürütülen işlemlerin önemli bir unsuru haline gelmiştir. Bulut Bilişim sayesinde, süper bilgisayarların dışında da karmaşık ve büyük ölçekli işlemler gerçekleştirilebilir hale gelmiş ve sınırlı kaynaklara sahip kullanıcıların ihtiyaçlarına çözüm sunulmuştur. Bu çözüm, anlık sensör verisi işleme veya anlık görüntü işleme gibi daha önce imkânsız olan bazı uygulamaları mümkün kılmıştır. Ancak bu yeni durum, beraberinde yeni problemleri de getirmiştir. Hesaplamaların ve işlemlerin buluta aktarılması ve geri döndürülmesi belirli bir zaman gerektirir. Bilgisayar ağlarının önemli problemlerinden biri olan gecikme (latency) bu bağlamda da karşımıza çıkmıştır. Bu sorunu çözmek için, Mobil Uç Bilişim geliştirilmiştir. Özellikle 5G teknolojileriyle uyumlu çalışması planlanan bu mimaride, hesaplamaların uzak merkezi bulut bilişim sunucularına aktarılması yerine, mobil uç sunucuların kullanıcılara daha yakın bir şekilde konumlandırılması yoluyla gecikme süresinin kısaltılması hedeflenmiştir. Ne yazık ki, sunduğu faydalara rağmen, Mobil Uç Bilişim yeni zorluklar yaratmıştır. Yaygın olarak kabul gören 4 ana zorluk vardır: (Rodrigues vd. 2020)

- Hesaplama Aktarım (Offloading Decision)
- Kaynak Tahsisi (Resource Allocation)
- Sunucu Konumlandırma (Server Deployment)
- Ek Yük Yönetimi (Overhead Management)

Bu zorluklar birbirine bağlıdır. Araştırma makalelerinde önerilen çözümler genellikle birden fazla zorluğu ele almaktadır. Genel sistem verimliliğini artırmak, genellikle birden fazla fayda sağlamaktadır. Dolayısıyla bir problemde elde edilmiş bir geliştirme diğer problemlerin çözümüne de katkı sunmaktadır.

### 1.1.1 Tez kapsamının sınırları ve odak noktası

Rodrigues vd. (2020) dört MEC zorluğunu (hesaplama aktarımı, kaynak tahsisi, sunucu konumlandırma, ek yük yönetimi) birbirine bağlı olarak tanımlamaktadır. Bu tez doğrudan hesaplama aktarım kararına odaklanmaktadır. Diğer üç zorluk Bölüm 2'de kuramsal çerçeve olarak sunulmuştur. Kaynak tahsisi, sunucu konumlandırma ve ek yük yönetimi ise Bölüm 3.4'teki  $F(x)$  (gecikme ve kaynak maliyetini birleştiren amaç fonksiyonu) alt bileşenleri aracılığıyla dolaylı olarak modele yansıtılmıştır. Örneğin “connection\_speed” ve “user\_density” ağ ve yoğunluk etkisini, “cpu\_power\_availability” ve “memory\_availability” kaynak kısıtını, “network\_queue” ise ek iletişim yükünü temsil etmektedir (Mao vd. 2017; Mach vd. 2017).

Çalışmanın sınırları şunlardır: (i) Tek görev–tek karar (yerel veya uç) modeli benimsenmiştir. Kısmi aktarım ve çoklu görev kuyruğu kapsam dışıdır (Mach vd. 2017). (ii) Karar, otuz iki boyutlu karar değişkeni vektörü  $x$  ile kodlanır. Amaç,  $F(x)$  değerini minimize etmektir (Bölüm 3.4). (iii) Beş meta-sezgisel algoritma (GA, PSO, GWO, ACO, ABC) aynı hiperparametre ve aynı değerlendirme fonksiyonu altında kıyaslanmıştır. Makine öğrenimi tabanlı yöntemler literatürde (Bölüm 2.6) karşılaştırma çerçevesi olarak konumlandırılmış, ancak bu tezde uygulanmamıştır (Mohammadi vd. 2018; Shakarami vd. 2020).

Bu sınırlar, Bölüm 1.3'te tanımlanan araştırma soruları (AS1–AS4) ve bilimsel katkılar (K1–K4) ile uyumludur. Tezin iddiası, “en iyi tek algoritma” değil, “bağlama göre seçilebilmektedir algoritma seti” ve bunu destekleyen simülatör–üretici–veri dönüşüm hattıdır.

### 1.1.2 Okuyucu için kavram haritası

Bu tezde tekrar eden kavramlar, okumanın geri kalanını kolaylaştıracaktır. Kısaltmalar: AS = araştırma sorusu (AS1–AS4, Bölüm 1.3.1), H = araştırma hedefi (H1–H5, Bölüm 1.3.2), K = bilimsel katkı (K1–K4, Bölüm 1.3.3). Temel kavramlar: (1) MEC — işlemin kullanıcıya yakın uç sunucuda yapılması (Mach vd. 2017) (Liu vd. 2020). (2) Hesaplama aktarımı — görevin telefonda mı, uçta mı çalışacağına karar vermek (You vd. 2017). (3) Görev (Task) vektörü  $x$  (Bölüm 3.1). (4) Amaç fonksiyonu  $F(x)$  — gecikme ve kaynak maliyetinin tek skorda birleşimi (Denklem 3.1); hedeflenen sonuç bu skorun olabildiğince küçük olmasıdır (Rao 2009) (Mao vd. 2017). (5) Meta-sezgisel algoritma  $F(x)$ 'i küçültmek için  $x$  uzayında arama yapan yöntem (GA, PSO, GWO, ACO, ABC).

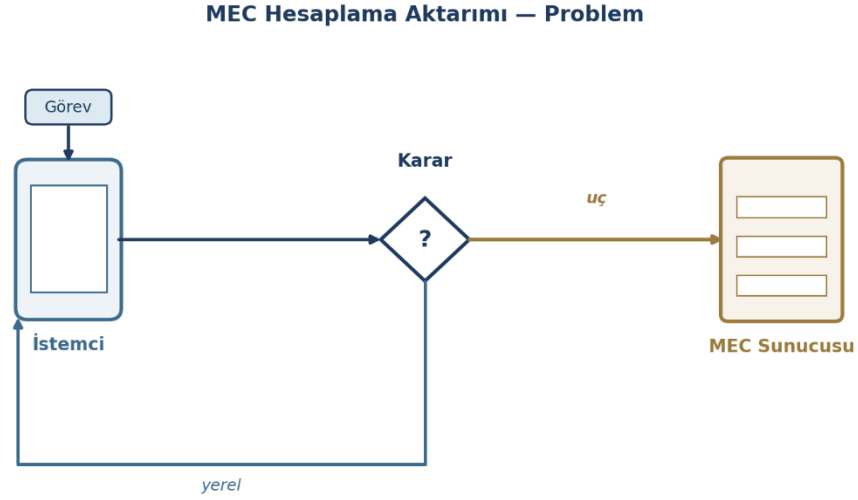
Akış özetle şöyledir: görev üretilir → algoritma  $x$  adaylarını dener → simülatör yerel ve uç maliyetini hesaplar → en düşük  $F(x)$  seçilir (Bölüm 1.2; Bölüm 3.4). Ayrıntılı karşılaştırma Bölüm 4'te yapılmaktadır.

Geliştirilen sistem ve veri simülatörünün diğer araştırmacılara yöntemsel referans sunabileceği ve farklı veri setleri üzerinde yürütülen meta-sezgisel deneylerin literatüre katkı sunabileceği değerlendirilmektedir. Problem tanımının ayrıntılı formülasyonu Bölüm 1.2'da verilmiştir.

## 1.2 Motivasyon ve Tez Kapsamı

Bu tezde ele alınan problem, şekil 1.1'de de gösterildiği üzere MEC ortamında bir görevin yerelde mi yoksa uç sunucuda mı çalıştırılacağına karar verilmesidir. Literatürde bu karar; kaynak tahsisi, sunucu yerleşimi ve ek yük ile birlikte düşünülür; ancak bu çalışmada

doğrudan odak hesaplama aktarımıdır (Rodrigues vd. 2020; Bölüm 1.1.1). Her deney örneği (görev), otuz iki boyutlu karar değişkeni vektörü  $x$  ile kodlanır. Simülatör aynı görev için yerel ve uç yolunu hesaplar; daha düşük toplam maliyet  $F(x)$  veren seçenek aranır (Bölüm 3.4).  $F(x)$ , gecikme ve kaynak kullanımını tek skorda birleştirir.



Şekil 1.1 Problem tanımı

Tez, her ortam için tek bir en iyi algoritma bulmaktan çok, aynı kurallar altında (aynı  $F(x)$ , aynı standart öznitelik şeması, aynı referans çizgisi) beş meta-sezgisel yöntemin (GA, PSO, GWO, ACO, ABC) farklı senaryo ve veri setlerinde nasıl sıralandığını incelemeyi ve buna dayalı bir seçim çerçevesi önermeyi hedefler (AS1–AS4). Kısmi aktarım, çoklu görev kuyruğu ve RL tabanlı politika öğrenimi bu tez kapsamı dışındadır.

Bulut bilişim yüksek kapasiteli işlemci gücünü her türlü kullanıcı için kullanılabilir hale getirmesi sebebiyle rağbet görmüştür. Karmaşık ve büyük işlemler süper bilgisayarlar dışında da yapılabilmesine ön ayak olup, kısıtlı kaynaklara sahip kullanıcılara alternatif bir çözüm sunmuştur. Ancak bu yeni durum kendi içinde yeni problemleri de getirmiştir. Hesaplama ve işlemlerin buluta aktarılması ve geri dönmesi belli bir zaman almaktadır. Bilgisayar ağlarının her aşamasında önemli yer tutan gecikme (delay) burada da karşımıza çıkmıştır. Bu problemi çözmek için Mobil Uç Bilişim ortaya atılmıştır. Özellikle 5G teknolojileriyle entegre çalışması hedeflenen bu mimaride uzaktaki merkezi bulut bilişim sunucularına hesaplama aktarımı yerine kullanıcılara daha yakın mobil uç sunucular konumlandırılarak arada geçen gecikme süresinin kısaltılması amaçlanmaktadır. Bir nevi

vekil sunucu görevi gören mobil uç bilişimde iki önemli süreç vardır. Bunlardan ilki sunucuların optimum konumlandırılması ikincisi hesaplama aktarımıdır. Bu tez, hesaplama aktarımında evrensel tek bir optimum yöntem aramak yerine, ortam profiline göre uygun meta-sezgisel yöntemi seçmeye yardımcı olacak bir modelleme ve karşılaştırma çerçevesi geliştirmeyi amaçlamaktadır. Geleneksel yaklaşımlar, mobil kullanıcı alışkanlıkları ve ağ parametreleri hakkında önceden bilgi sahibi olmayı gerektirir. Ancak MEC (Mobil Uç Bilişim) çok dinamik koşullara sahiptir ve mobil yapısı nedeniyle statik veya geleneksel ağ çözümleri istenen etkiyi gösterememektedir. Geleneksel ağ yapısında bileşenler statiktir. Ağın boyutu ne kadar büyük olursa olsun, sabit doğası araştırmacıların topografyadan kurallar ve desenler çıkarmasını, dinamik ağ yapılarına kıyasla daha kolay hale getirir. Bu noktadan sonra algoritma ve protokol geliştirme süreci basitleşir çünkü varsayımlar iyi bir şekilde tanımlanmış ve stabildir. Bu nedenle, önerilen çözümler büyük ölçekli sistemlerde uygulanabilir. Uzun yıllardır yapılan çalışmalar çözümleri oldukça rafine etmiş, çok verimli hale getirmiştir.

Ancak MEC için stratejiler geliştirmeye gelince, bu sağlam temeller sarsılmaya başlamaktadır. Topografya hem terminallerin hem de uç sunucuların hareketli olması nedeniyle "akışkan" bir yapıya dönüşür. Buna ek olarak, günümüzde terminaller hem daha yetenekli hem de çeşitlidir. Bu durumun dikkate alınması gereklidir. Bir diğer değişken faktör ise sunucu kapasitesidir. Bu bile sabit değildir. Bu nedenle, aşağıda sıralanan nedenlerden dolayı dinamik çözümler gereklidir (Cao vd. 2019):

- Mobil kullanıcının kalan iş yükü, yerel işlemler, yeni eklenen görevler, zaman aşımına uğramış veya kaldırılmış görevler nedeniyle değişkendir.
- Hedef MEC sunucusunun mevcut bant genişliği ve kapasitesi, diğer kullanıcıların kararları ve eylemleri nedeniyle değişkendir.
- Hedef MEC sunucusuna bağlı mobil kullanıcı sayısı değişkendir.
- Mobil kullanıcının hareketli yapısı ve ağ ayarlarının değişebilmesi nedeniyle hedef MEC sunucusu da değişkendir.

Yukarıdaki nedenler ışığında makine öğrenimi (ML) ve dinamik optimizasyon yöntemleri öne çıkmaktadır (Rodrigues vd. 2020) (Mohammadi vd. 2018) (Cao vd. 2019).

ML yöntemleri, gizli bilgi ve desenleri çıkarmada etkili olduğu raporlanmıştır. Modern ML çerçevelerinin ortaya çıkmasından bu yana optimizasyon için yoğun bir şekilde kullanılmaktadır. Araştırmacılar MEC ve ML üzerine yoğun çalışmalar yapmış ve kayda değer bulgular raporlanmıştır. Bu çalışma farklı bir doğrultuda MEC bağlamında daha az çalışılmış olan meta-sezgisel algoritmalara ağırlık vermiştir.

### 1.2.1 Meta-sezgisel yaklaşımın gerekçesi

MEC ortamında RL/DRL yaygın olmasının nedeni, dinamik durum uzayında politika öğrenimidir (Zhang vd. 2019; Wang vd. 2019a; Chen vd. 2019). Buna karşılık meta-sezgisel yöntemler, açık  $F(x)$  tanımı olduğunda üç avantaj sunmaktadır: (1) Karar mantığı simülör formülleriyle doğrudan izlenebilir (Rao 2009). (2) Senaryo veya veri seti değişince aynı kod tabanı yeniden koşulabilmektedir. Uzun eğitim döngüsü zorunlu değildir (Mohammadi vd. 2018). (3) Çizelge 2.1'de Ale vd. (2019) ve Dong vd. (2019) örneklerinde görüldüğü üzere, sürü zekâsı tabanlı yöntemler aktarım literatüründe yer bulmaktadır. Bu tez bu çizgiyi beş algoritma ve on beş senaryo ile sistemleştirmektedir.

Sezgisel tercih, “daha az çalışılmış” iddiasını mutlak değil görelî olarak kullanmaktadır. RL makale sayısı daha fazladır. Aynı  $F(x)$ , aynı standart öznitelik şeması ve aynı referans çizgisi (baseline) altında GA–PSO–GWO–ACO–ABC'nin birlikte raporlandığı çalışma sayısı sınırlıdır (Shakarami vd. 2020). Bu boşluk, Bölüm 1.3.2'de tanımlanan H2 (beş algoritma kıyası) ve Bölüm 1.3.3'te tanımlanan K2 (senaryo-şartlı üstünlük kanıtı) hedeflerinin temelini oluşturmaktadır.

### 1.2.2 MEC bağlamında meta-sezgisel yaklaşım

Tezin özü, Multi-Access Edge Computing (MEC) ortamında hesaplama aktarım kararının meta-sezgisel algoritmalarla nasıl modellenebileceği, neden bu araçların uygun olduğu ve hangi koşullarda hangi faydayı sağlayabileceğidir. Bu üç soru, sonraki bölümlerdeki deneysel tasarımın kuramsal temelidir (Mach vd. 2017) (Mao vd. 2017).

Neden: Aktarım kararı; görev boyutu, mesafe, bağlantı hızı, kullanıcı yoğunluğu ve sunucu kapasitesi gibi çok sayıda değişkenle birlikte değerlendirildiğinde kombinatorial ve dinamik bir yapıdadır (Cao vd. 2019; Rodrigues vd. 2020). Literatürde takviyeli öğrenme yaygın olsa da (Zhang vd. 2019; Wang vd. 2019a), etiketli karar dizileri ve uzun eğitim süreçleri her operatör senaryosu için her zaman mümkün değildir (Mohammadi vd. 2018; Shakarami vd. 2020). Meta-sezgisel yöntemler, açık tanımlı bir amaç fonksiyonu  $F(x)$  verildiğinde türev bilgisi gerektirmeden arama yapabilir (Rao 2009). Çizelge 2.1'deki sürü zekâsı ve sezgisel çalışmalar (Ale vd. 2019; Dong vd. 2019; Vimal vd. 2020), bu çizginin aktarım literatüründe yer bulduğunu göstermektedir. Bu tez aynı problemi beş algoritma ve ortak protokolle sistemleştirmektedir.

Nasıl: Karar modeli ve  $F(x)$  değerlendirmesi Bölüm 1.2 ve Bölüm 3.4'te açıklanmıştır. GA, PSO, GWO, ACO ve ABC aynı popülasyon ve iterasyon ayarlarıyla  $x$  uzayında arama yapmaktadır. Fark operatörlerde oluşmaktadır: elitizm ve çaprazlama (GA) (Goldberg 1989), sürü hızı ve gbest (PSO) (Kennedy vd. 1995), kurt hiyerarşisi (GWO) (Mirjalili vd. 2014), feromon (ACO) (Dorigo vd. 2006), limit ve koloni rolleri (ABC) (Karaboga vd. 2007). Senaryo adı yalnızca başlangıç dağılımını değiştirir; algoritma çekirdeği aynı kalmaktadır (Bölüm 3.1.2).

Beklenen fayda: (i) Gecikme ile kaynak kullanımı arasında senaryoya göre dengelenmiş karar (Mach vd. 2017). (ii) Yüksek yoğunluklu profillerde PSO/GWO ile daha düşük gecikme potansiyeli. Düşük kaynak profillerde GA/ABC ile daha dengeli kaynak tüketimi (Bölüm 4.1). (iii) Karar mantığının simülatör formülleriyle izlenebilir olması (Rao 2009). (iv) Senaryo değiştikçe aynı yazılımın yeniden çalıştırılabilmesi. Bu özellik, RL tabanlı çözümlerde sık görülen yeniden eğitim maliyetine kıyasla karşılaştırma hızı sağlamaktadır (Mohammadi vd. 2018).

### 1.3 Tezin Yapısı

Tez çalışmasının devamı şu şekilde düzenlenmiştir:

- ✓ Bölüm 2’de çalışma kapsamında incelenmiş IoT, Bulut Bilişim, MEC üzerine yapılmış çalışmaların özetlerinin ve çıkarımların bulunduğu literatür özeti sunulmuştur. Literatür taraması sonucunda ulaşılan sonuçların ve yöntemlerin genel bir bakış açısına göre değerlendirmesi paylaşılmıştır. Bu bölümde yöntem tercihlerinin altında yatan sebepler de açığa çıkmıştır.
- ✓ Bölüm 3’te geliştirilen sistemin tüm detayları ayrıntılı biçimde anlatılmıştır. Veri oluşturucu, simülasyon mimarisi ve uygulanan algoritmalar çeşitli çizelgelerle desteklenmiştir. Ayrıca değerlendirme kriterlerine ve uygulama senaryolarına da bu bölümde yer verilmiştir.
- ✓ Son olarak, 4. bölüm farklı senaryolar için gerçekleştirilen simülasyon testlerinin sonuçlarını, bu sonuçların yorumlarını ve gelecekteki potansiyel hedefleri içermektedir.

### 1.3.1 Araştırma soruları

Aşağıdaki araştırma soruları (AS), Bölüm 1.2’deki problem tanımına yanıt aramaktadır:

AS1 — Tek bir meta-sezgisel algoritma, farklı MEC senaryo profillerinde (düşük/yüksek kaynak, statik/dinamik) hem gecikme hem kaynak kullanımı açısından sürekli üstün müdür?

AS2 — GA, PSO, GWO, ACO ve ABC’nin göreceli üstünlükleri, algoritma operatörleri (elitizm, sürü keşfi, feromon güncelleme vb.) ile nasıl ilişkilendirilebilir ve literatürle uyumlu mudur? (Goldberg 1989) (Kennedy vd. 1995) (Mirjalili vd. 2014) (Dorigo vd. 2006) (Karaboga vd. 2007)

AS3 — Ortak bir amaç fonksiyonu  $F(x)$  ve yerel/uç simülasyonu altında, sentetik Poisson senaryoları ile gerçek dünya veri setlerinden türetilen görev havuzları tutarlı karşılaştırma zemini sağlamakta mıdır? (Guo vd. 2019) (Hadary vd. 2020)

AS4 — Senaryo ve veri seti profiline göre algoritma seçimi için uygulanabilir bir karar

çerçevesi önerilebilir mi?

### 1.3.2 Araştırma hedefleri

Bu sorulara yanıt vermek için hedefler şöyle tanımlanmıştır: (H1) Otuz iki boyutlu standart öznitelik şeması ve Poisson tabanlı on beş senaryo ile tekrarlanabilir bir simülasyon ortamı kurmak. (H2) Beş meta-sezgisel algoritmayı aynı hiperparametre ve referans çizgisi (tümü yerel / tümü kenar) politikasıyla kıyaslamak. (H3) Gecikme ve kaynak metriklerini yalnızca simülatör formülleriyle hesaplayarak yöntemler arası adaleti sağlamak (Bölüm 3.4.2). (H4) Ayrıca gerçek dünya veri setlerinden görev havuzu üreterek dış geçerliliği desteklemek (Bölüm 3.5.3–3.5.8, 4.1.3). (H5) Bulguları senaryo/veri seti tipine göre algoritma seçim önerilerine dönüştürmek (Bölüm 4.1.16).

### 1.3.3 Bilimsel katkılar

Çalışmanın bilimsel katkıları dört başlıkta özetlenmektedir:

(K1) MEC hesaplama aktarımı için otuz iki değişkenli, yerel/uç ikili değerlendirmeli açık formüllü simülatör ve veri üretici.

(K2) Aynı  $F(x)$  altında beş meta-sezgiselin on beş sistematik senaryoda karşılaştırılması ve senaryo-şartlı üstünlük kanıtı.

(K3) Gerçek dünya veri setlerinin standart öznitelik şemasına projeksiyonu ve veri seti koşullu deney hattı (adaptör, kalibrasyon, hibrit).

(K4) Düşük kaynaklı ortamlarda GA/ABC, yüksek yoğunluklu ortamlarda PSO/GWO eğilimini destekleyen, literatürle hizalı seçim çerçevesi.

İlk olarak (K1), MEC ortamlarında hesaplama aktarımı probleminin analizine yönelik olarak, otuz iki değişkeni dikkate alan ve yürütme kararlarını modelleyebilen açık formüllü bir simülatör ve veri üretim altyapısı geliştirilmiştir. İkinci olarak (K2), aynı amaç fonksiyonu altında beş farklı meta-sezgisel algoritmanın on beş sistematik senaryoda performansları karşılaştırılmış ve algoritmaların farklı koşullar altındaki göreceli üstünlükleri ortaya konulmuştur. Üçüncü olarak (K3), gerçek dünya veri setlerinin önerilen standart öznitelik şemasına şeffaf biçimde dönüştürülmesini sağlayan bir veri işleme çerçevesi oluşturulmuştur. Bu kapsamda adaptör, kalibrasyon ve hibrit eşleme yöntemleri kullanılarak veri seti koşullu deneysel bir değerlendirme hattı geliştirilmiştir. Son olarak (K4), elde edilen deneysel bulgular doğrultusunda, düşük kaynaklı ortamlarda Genetik Algoritma (GA) ve Yapay Arı Kolonisi (ABC), yüksek yoğunluklu ortamlarda ise Parçacık Sürü Optimizasyonu (PSO) ve Gri Kurt Optimizasyonu (GWO) algoritmalarının daha avantajlı olduğunu gösteren ve literatürdeki bulgularla uyumlu bir algoritma seçim çerçevesi önerilmiştir.

#### **1.3.4 Tez bölümlerine genel bakış**

Bölüm 2, kuramsal zemini ve literatür boşluğunu (ML/DRL baskın, sezgisel karşılaştırmanın sınırlı kapsamı, veri eksikliği) ortaya koymaktadır. Bölüm 2.6.2–2.6.3 gerçek dünya veri setleri ve simülatör tercihini gerektirmektedir. Bölüm 3, yöntemi uçtan uca tanımlamaktadır: veri üretimi (3.1–3.1.2), mimari (3.2), algoritmalar (3.3),  $F(x)$  (3.4), senaryolar ve referans çizgisi (3.5–3.5.2), gerçek dünya veri dönüşüm hattı ve tekrarlanabilirlik (3.5.3–3.5.8). Bölüm 4.1 Poisson senaryo ve gerçek dünya veri seti sonuçlarını sunmaktadır. Bölüm 4.1.14 iki eksenli çapraz kıyaslamaktadır. Bölüm 4.1.9  $n \geq 10$  seed tekrarlı koşu ve Bölüm 4.1.11 Wilcoxon raporlama protokolünü tanımlamaktadır. Bölüm 4.1.16 bulguları sentezlemektedir. Bölüm 4.2.6 sınırlamaları ve gelecek çalışmaları listelemektedir.

### 1.3.5 Deneysel kapsam ve metodolojik bütünlük

Bu tez, hesaplama aktarımı problemine iki tamamlayıcı deneysel eksen den yaklaşmaktadır. Birinci eksen, Poisson tabanlı on beş senaryo, veri üretici, simülasyon mimarisi, beş sezgisel algoritma, amaç fonksiyonu ve kontrollü referans çizgisi karşılaştırmasını kapsamaktadır (Bölüm 3.1–3.5). İkinci eksen, Alibaba, Azure, GeoLife, 5G-MoNArch ve MLPerf gibi gerçek dünya veri setlerinin adaptör, kalibrasyon ve hibrit birleşim yöntemleriyle standart öznitelik şemasına projeksiyonunu ve aynı optimizasyon çekirdeği altındaki değerlendirmeyi kapsamaktadır (Bölüm 2.6.2–2.6.3, 3.5.3–3.5.8, 4.1.3). Her iki eksen de aynı  $F(x)$ , aynı popülasyon/iterasyon limiti ve aynı raporlama ilkeleri geçerlidir. Metrikler simülatör formüllerinden türetilmektedir (Mohammadi vd. 2018; Shakarami vd. 2020).

## **2. KAYNAK ÖZETLERİ VE KURAMSAL TEMELLER**

Tez çalışmasının bu bölümünde, araştırmanın bağlamını oluşturan Mobil Uç Bilişim, Nesnelerin İnterneti ve Bulut Bilişim kavramları üzerine kapsamlı bilgiler sunulmuştur. Öncelikle bu teknolojilerin temel tanımları, çalışma prensipleri ve birbiriyle olan ilişkileri açıklanmış; ardından bu alanlarda karşılaşılan teknik zorluklar ve performans kısıtları üzerinde durulmuştur. Çalışma kapsamında çözüm üretilen problemlerin tanımları ayrıntılı olarak yapılmıştır. Ayrıca literatürde yer alan güncel akademik ve endüstriyel çalışmaların bulguları karşılaştırmalı olarak değerlendirilmiştir. Böylece, çalışmanın motivasyonu, literatürdeki boşluklar ve önerilen çözümün hangi gereksinimlerden doğduğu ortaya konulmuştur.

### **2.1 Nesnelerin İnterneti ve Bulut Bilişim**

Kapsamı yeterince ele almak için, MEC'in (Mobil Uç Bilişim) zorluklarına geçmeden önce, Nesnelerin İnterneti (IoT) ve Bulut Bilişim'in mevcut eğilimlerine hızlı bir genel bakış sunmak gerekmektedir. Nesnelerin İnterneti (IoT), fiziksel cihazların birbirine bağlandığı ve veri ile bilgi topladığı ve paylaştığı bir ağdır. Genellikle sensörler, yazılımlar ve diğer teknolojilerle donatılmış olan bu cihazlar, internet veya diğer iletişim ağları üzerinden birbiriyle ve diğer sistemlerle iletişim kurabilir ve etkileşime girebilir.

Bulut Bilişim, internet üzerinden sunucular, depolama, uygulamalar ve hizmetler gibi yapılandırılabilir bilişim kaynaklarına talep üzerine erişim sağlayan bir teknolojidir. Bu teknoloji, kullanıcıların uzak veri merkezlerinde veri depolamalarını, yönetmelerini ve işlemelerini mümkün kılarak ölçeklenebilirlik, esneklik ve maliyet verimliliği sunmaktadır. Bulut Bilişim ile Nesnelerin İnterneti'nin entegrasyonu, bulutun ölçeklenebilirliği ve işlem gücünden yararlanarak IoT uygulamalarının yeteneklerini artırmayı amaçlayan son dönem araştırmalarında önemli bir odak noktası olmuştur. Bu entegrasyon, farklı zorluklara ve gereksinimlere yanıt veren çeşitli platformlar ve mimariler geliştirilmesine yol açmış ve doğal olarak birçok araştırmacının ilgisini çekmiştir.

Ishaq ve arkadaşları, bu zorlukları makalelerinde detaylı bir şekilde ele almıştır. Makale, IoT ve Bulut Bilişim'in birleşiminde ortaya çıkan güvenilirlik, güvenlik, veri depolama ve erişim, hesaplama performansı ve depolama, uç bilişim, bakım, cihaz etkileşimi ve kullanıcı destekli cihazlar gibi çeşitli zorluklara odaklanmaktadır. Her bir zorluk, IoT-Bulut Bilişim paradigması üzerindeki etkilerini ortaya koymak için ayrıntılı olarak incelenmiştir. Örneğin, IoT cihazlarının zamana duyarlı uygulamalar için buluta bağımlılığı, güvenilirlik konusunda endişeler doğururken, kritik verilerin güvenli bir şekilde buluta iletilmesi ihtiyacı, güvenlik açısından önemli bir sorun olarak vurgulanmıştır (Ishaq vd. 2021).

Makine Öğrenimi (ML) ve Yapay Zeka (AI) bu zorluklar için önerilmiştir. Rath ve arkadaşları, AI ve ML algoritmalarının IoT sistemlerinde ve bulut bilişim ortamlarında yer alan karmaşık sorunları çözmek için nasıl uygulandığını gösteren birçok vaka çalışması ve araştırma örneği sunmuştur. Yazarların odak noktalarından biri, veri yönetimi, kaynak tahsisi, sunucu yerleştirme ve ek yük yönetimi gibi konularda AI ve ML'nin kritik rolüdür. Denetimli ve denetimsiz öğrenme, takviyeli öğrenme ve derin öğrenme gibi makine öğrenimi algoritmalarının, IoT cihazlarının ve bulut hizmetlerinin performansını ve güvenliğini optimize etmedeki önemini vurgulamışlardır (Rath vd. 2021).

2020 yılında Alhaidari ve arkadaşları, IoT ve Bulut Bilişim'in entegrasyonu için "Nesnelerin Bulutu" (Cloud of Things, CoT) terimini önermiştir. Benzer şekilde, CoT'un enerji verimliliği, kaynak tahsisi, kimlik yönetimi, hizmet keşfi ve güvenlik gibi zorluklarını vurgulamışlardır. Bu zorluklara çözüm olarak gelişmiş ağ teknolojileri, verimli veri işleme teknikleri ve sağlam şifreleme mekanizmalarının önemine dikkat çekmişlerdir. Makale, CoT'un IoT sistemlerinin etkinliğini ve uygulanabilirliğini artırmadaki önemini vurgulamakta ve belirlenen zorlukların aşılmasının daha ölçeklenebilir, güvenli ve verimli IoT uygulamalarına yol açacağını öne sürmektedir (Alhaidari vd. 2023).

Jahantighi ve arkadaşları, IoT ve Bulut Bilişim'in entegrasyonunun sistematik bir analizini sunmuştur. Araştırmayı üç ana alana ayırmışlardır: uygulamalar, platformlar

ve diğerk teknolojilerle entegrasyon. Makale, IoT cihazlarının hızlı büyümesi ve ölçeklenebilir bulut hizmetlerine duyulan ihtiyaca dikkat çekmiş, uç bilişimin uç sistemlere daha yakın hizmet sunma konusundaki kritik rolünü vurgulamıştır. Yazarlar, mobil bulut bilişim, güvenlik sorunları ve IoT cihazlarının işlem gücü ve depolama ihtiyaçlarını da tartışmışlardır (Hamdan vd. 2020).

Singh ve arkadaşları da IoT ve Bulut Bilişim entegrasyonunun potansiyel faydalarını vurgulamış, özellikle sağlık alanındaki uygulamalara odaklanarak bu eğilimlerin pratik çözümlerini göstermiştir (Singh vd. 2021). MEC (Çok Erişimli Uç Bilişim) de IoT ve Bulut Bilişim entegrasyonunda çok kritik bir rol oynamaktadır. Liu ve arkadaşları, MEC'in 5G ve IoT teknolojileriyle entegrasyonunu ele almış ve MEC'in gecikmeyi azaltma ve bant genişliğini iyileştirme gibi avantajlarını vurgulamıştır. Bu, akıllı şehirler ve AR/VR gibi gelişen IoT uygulamaları için kritik öneme sahiptir. Makale, MEC'i mümkün kılan anahtar teknolojileri (bulut bilişim, yazılım tanımlı ağlar, ağ işlev sanallaştırması) ve gelecekteki araştırma yönelimlerini tartışmaktadır (Liu vd. 2020).

## **2.2 Hesaplama Aktarım**

Hesaplama aktarımı, Mobil Uç Bilişim'in (MEC) temel bileşenlerinden biridir ve hesaplama görevlerinin mobil cihazlardan uç sunuculara aktarılmasını sağlamaktadır. MEC'de hesaplama aktarımının birincil amacı, gecikmeyi azaltmak ve mobil cihazların genel performansını artırmaktır. Hesaplama aktarımının temel prensibi, mobil cihazlardaki hesaplama yükünü hafifletmek için görevleri, daha üstün kaynaklara sahip ve/veya cihazlara daha yakın olan uç sunuculara aktarmaya dayanmaktadır.

Hesaplama aktarımı, MEC'in muhtemelen en kritik ve karmaşık yönüdür ve MEC mimarisini önemli ölçüde geliştirme potansiyeline sahiptir. Bu nedenle, bu alandaki araştırmaların büyük bir kısmı, verimli aktarım stratejileri geliştirmeye odaklanmıştır. MEC'in endüstriyel alanlardan kentsel çevrelere kadar çeşitli alanlarda uygulanabilirliği göz önüne alındığında, araştırmacılar bu konseptin farklı ortam koşullarındaki faydalarını incelemektedir (Liao vd. 2019) (Zhou vd. 2020) (Zhang

vd. 2019). Baęlamdan baęımsız olarak, aktarım kararı MEC mimarisinin merkezi bir unsuru olarak kalmakta ve akademik evrelerde nemli bir arařtırma konusu olmaktadır.

Hesaplama aktarım sreci, grevlerin veya iřlemlerin bir u sunucuya aktarılmasını iermektedir ve bu sayede aęa baęlı cihazların iřlemleri devretmesi saęlanmaktadır. Bulut Biliřim'den farklı olarak, cihazların grevleri sınırlı kapasiteleri nedeniyle devrettięi durumların aksine, MEC'de aktarım genellikle gecikme sorunlarını azaltmak iin kullanılmaktadır. Bu durum, MEC'nin mobil bir ortamda deęiřken kořullarla alıřtıęı gz nne alındıęında aktarım srecine karmařıklık katmaktadır. Dolayısıyla, aynı grevin aktarımı tutarlı yrtme srelerini azaltmayı hedeflemektedir. Bu sorunu zmek iin cihazların adaptif karar mekanizmalarına ihtiyaı bulunmaktadır.

Farklı kullanım senaryolarında  olası sonu belirlenmiřtir: Yerel Yrtme (Local Execution), Tam Aktarım (Full Offloading) ve Kısmi Aktarım (Partial Offloading) (Mach vd. 2017). Makine ęrenimi (ML) algoritmaları, aę kořulları, cihaz yetenekleri ve uygulama gereksinimleri gibi faktrleri analiz edebilme yetenekleri nedeniyle aktarım karar problemini zmek iin zellikle uygundur. Bu teknikler, hangi grevlerin aktarılacaęı ve nereye aktarılacaęı konusunda akıllı kararlar verebilmek iin eřitli faktrleri analiz edebilir. Yapay Zek (AI) kullanarak, U Biliřim ortamlarındaki hesaplama aktarımı daha verimli, uyarlanabilir ve evredeki dinamik deęiřimlere duyarlı hale getirilebilir (Carvalho vd. 2020).

### **2.2.1 IoT–Bulut–MEC zinciri ve hesaplama aktarımına geiř**

Blm 2.1'de zetlenen IoT ve Bulut Biliřim entegrasyonu (Ishaq vd. 2021) (Rath vd. 2021) (Hamdan vd. 2020), kaynak kısıtlı cihazların merkezi iřlem gcne eriřimini mmkn kılmaktadır. Liu vd. (2020), MEC'in 5G ve IoT ile birleřerek gecikmeyi azalttıęını vurgulamaktadır; akıllı řehir ve AR/VR gibi uygulamalar iin bu kritiktir. Bu zincirde hesaplama aktarımı, “grevin nerede yrtleceęi” sorusunu merkeze almaktadır (Mach vd. 2017) (You vd. 2017).

Tez, bu geiři řu üç bileřenle sınırlandırmaktadır: (i) karar deėiřkeni olarak otuz iki boyutlu karar deėiřkeni vektörü  $x$  (ii) ama fonksiyonu olarak  $F(x)$ 'in minimize edilmesi (iii) yerel ve u yürütme yollarının simülatör skorlarıyla deėerlendirilmesi (Bölüm 3.4). Literatürde RL/DRL baskın olsa da (Zhang vd. 2019) (Wang vd. 2019a), açık  $F(x)$  tanımı altında yürütölen sezgisel kıyaslama, yorumlanabilirlik aısından avantaj sunmaktadır (Ale vd. 2019) (Shakarami vd. 2020). Ařaėıdaki alt bölümlerde kaynak tahsisi, sunucu konumlandırma ve ek yük yönetimi ele alınmaktadır. Ancak deneysel odak ise hesaplama aktarım kararı olarak belirlenmiřtir...

Sonuç olarak, MEC'in en önemli avantajı, gecikmeyi azaltmasıdır. Hamdan ve arkadaşları, verilerin kaynaėa daha yakın iřlenip depolanmasını saėlayarak gecikmeyi önemli ölçüde azalttıėını ve IoT uygulamalarının verimliliėini artırdıėını vurgulamıřtır. Ayrıca, u biliřim mimarilerini mevcut IoT katman modelleriyle eřleřtirerek her mimarinin yeteneklerini ve boşluklarını haritalamıřlardır. Mevcut sınırlamalar ve gelecekteki arařtırma yönelimleri üzerine bir tartıřma ile sonuçlanan alıřma, u biliřim yoluyla IoT uygulamalarını geliřtirmek için ok yönlü bir yaklařım önermektedir (Hamdan vd. 2020).

### 2.2.2 Hesaplama aktarımında sezgisel yöntemlerin konumu

Hesaplama aktarım literatüründe takviyeli öėrenme ve derin öėrenme baskındır (Zhang vd. 2019) (Wang vd. 2019a) (Zhao vd. 2020). Bunun yanında denetimli/ denetimsiz öėrenme ve sürü zekâsı tabanlı sezgisel yöntemler de izelge 2.1'de özetlendiėi üzere yer bulmaktadır (Ale vd. 2019) (Dong vd. 2019) (Vimal vd. 2020). Bu tez, özellikle açık formüllü ama fonksiyonu (Denklem 3.1) ve ikili deėerlendirme ile tanımlanan problemde, yorumlanabilir ve tekrarlanabilir karřılařtırma için meta-sezgisel yaklařımı seçmiřtir (Goldberg 1989) (Rao 2009).

Sezgisel algoritmaların MEC'deki uygunluėu üç nedene dayanmaktadır: (1) Problem NP-zor sınıfına yakın kombinasyonel yapıdadır. Kesin özüm ok maliyetlidir (Taleb vd. 2017). (2) Simülatör tabanlı  $F(x)$  türevlenebilir olmak zorunda deėildir. Sürü ve koloni yöntemleri bu yüzden uygundur (Kennedy vd. 1995) (Dorigo vd. 2006). (3) Senaryo

değiştikçe aynı kod tabanı yeni  $\lambda$  veya yeni görev havuzu ile yeniden koşulabilmektedir. Bu, RL için gereken eğitim döngüsü maliyetini azaltmaktadır (Mohammadi vd. 2018).

Rodrigues vd. (2020) dört MEC zorluğunu listelemektedir. Bu çalışma hesaplama aktarımına odaklanmaktadır ancak kaynak tahsisi ve ek yük formüllerinde dolaylı olarak diğer zorluklara değinmektedir (Bölüm 3.4). Shakarami vd. (2020) makine öğrenimi tabanlı aktarım anketinde veri eksikliğini vurgulamaktadır. Veri üretici ve görev adaptörü bu boşluğu doldurma stratejisi olarak konumlandırılmıştır.

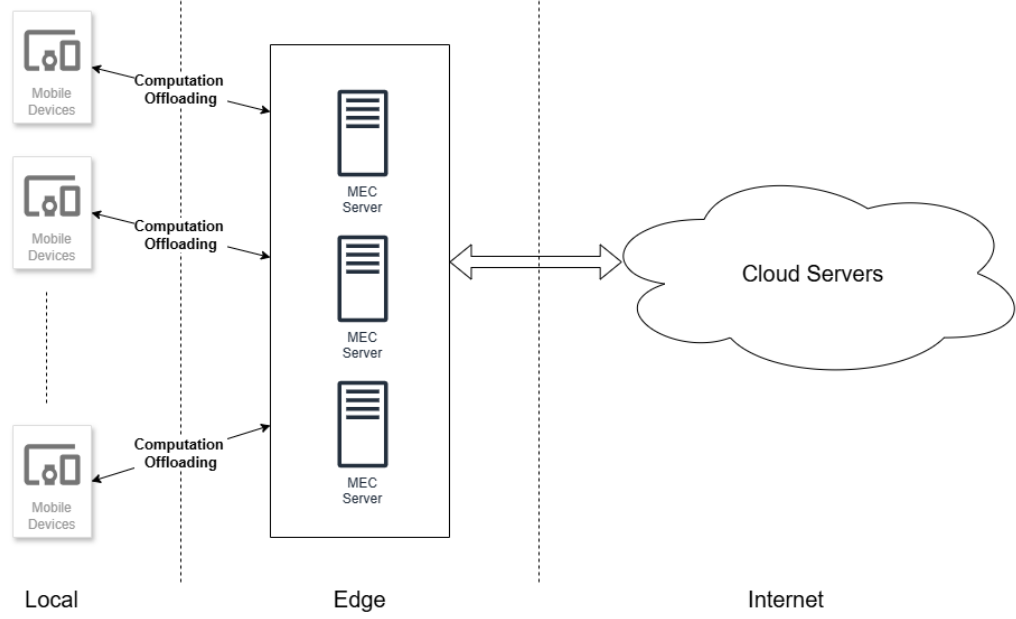
### **2.2.3 Aktarım kararında meta-sezgisellerin rolü ve RL ile konumlandırma**

Hesaplama aktarımında üç yürütme biçimi tanımlanmıştır: yerel, tam aktarım ve kısmi aktarım (Mach vd. 2017). Bu tez, karar uzayını yerel ile uç arasında ikili seçime indirgemektedir; amaç gecikme ve kaynak bileşenlerinin ağırlıklı birleşimini minimize etmektir (Bölüm 3.4). RL/DRL tabanlı çalışmalar, çevreyle etkileşimden öğrenilen bir karar mekanizmasını zaman içinde iyileştirerek uzun vadeli ödülü en üst düzeye çıkarmayı hedeflemektedir (Zhang vd. 2019) (Wang vd. 2019a). Meta-sezgisel çalışmalar ise her bir görev örneği için, önceden tanımlı iterasyon limiti dahilinde en uygun karar vektörünü ( $x$ ) aramaktadır (Goldberg 1989) (Kennedy vd. 1995). Bu yürütme biçimleri Şekil 2.1'de şematize edilmektedir.

İki yaklaşım rakip değil, tamamlayıcıdır: RL sürekli öğrenen sistemler için uygundur. Meta-sezgiseller simülatör tanımlandığında hızlı karşılaştırma ve hiperparametre analizi sağlamaktadır (Mohammadi vd. 2018). Ale vd. (2019) ve Dong vd. (2019), derin öğrenme ile sürü/sezgisel yöntemlerin aktarımda kullanıldığını göstermektedir. Vimal vd. (2020) oyun teorisi ve sezgisel zekâyı bir arada ele almaktadır. Bu tez, RL'yi uygulamadan önce hangi meta-sezgisel ailenin hangi MEC profiline daha uygun olduğu sorusuna yanıt aramaktadır (Bölüm 4.1.16).

Pratik düzeyde özetlenecek olursa, yüksek bant genişliği ve yoğun kullanıcı koşullarında keşif ağırlıklı PSO ve GWO algoritmalarının gecikmeyi düşürme potansiyeli taşıdığı;

kaynak kısıtlı ortamlarda ise GA ve ABC'nin kaynak verimliliği sağladığı görülmektedir. Orta düzey kaynak profillerinde ACO'nun feromon güncelleme maliyeti nedeniyle daha dikkatli değerlendirilmesi gerektiği de belirtilmelidir (Dorigo vd. 2006). Bu çıkarımlar, Bölüm 4'te senaryo bazında sunulan deneysel bulgularla desteklenmektedir.



Şekil 2.1 Hesaplama aktarım

#### 2.2.4 Yerel, tam ve kısmi aktarımın bağlamdaki yeri

Mach vd. (2017) hesaplama yürütmesini yerel, tam aktarım ve kısmi aktarım olarak sınıflandırmaktadır. Bu tezde yalnızca yerel (LOCAL) ve uç (EDGE) yolları değerlendirilmektedir; kısmi aktarım kapsam dışıdır (Bölüm 1.1.1). Her  $x$  için simülör iki yolu kıyaslayarak daha düşük  $F(x)$  üreten yolu seçer (Mao vd. 2017). Böylece aktarım kararı,  $x$  boyutlarının  $F(x)$  altında toplanmasından türeyen toplam maliyet karşılaştırmasıdır.

Takviyeli Öğrenme (Reinforcement Learning, RL) ve Derin Takviyeli Öğrenme (Deep Reinforcement Learning, DRL), sınırlı veri kümeleriyle çalışabilmeleri ve önceki deneyimlerden öğrenebilmeleri nedeniyle ML algoritmaları arasında tercih edilmektedir

(Zhang vd. 2019) (Wang vd. 2019a) (Zhao vd. 2020) (Chen vd. 2019) (Huang vd. 2018) (Zhang vd. 2019) (Vimal vd. 2020) (Xiao vd. 2018) (Feng vd. 2020) (Cheng vd. 2019) (Alam vd. 2019) (Gao vd. 2024) (Chen vd. 2023) (Liao vd. 2023). Diğer ML yaklaşımları arasında Denetimli Öğrenme (Supervised Learning), Denetimsiz Öğrenme (Unsupervised Learning), Derin Öğrenme (Deep Learning) ve Oyun Teorisi (Game Theory) de son çalışmalarda kullanılmaktadır (Liao vd. 2019) (Vimal vd. 2020) (Hao vd. 2018) (Ale vd. 2019) (Dong vd. 2019) (Huang vd. 2023) (Zaman vd. 2023) (Abdullaev vd. 2023) (Jang vd. 2023). Parametre Tabanlı Bağlam Farkındalığı (Parameter-Based Context Awareness), aktarım karar problemine uygulanan bir başka stratejidir (Farahbakhsh vd. 2023).

Araştırmaların en büyük kısmı bu problem üzerine yoğunlaşmaktadır. Bu durum, aktarım kararının MEC için en zorlu sorun olduğunu ve en fazla iyileştirme alanına sahip olduğuna işaret etmektedir. Özetle, hesaplama aktarımı MEC'in başarısı için kritik öneme sahiptir, çünkü mobil cihaz performansını artırmaktadır ve gecikmeyi azaltmaktadır. Aktarım kararlarının, kaynak tahsisinin ve ilişkili ek yüklerin verimli yönetimi, MEC'in sorunsuz çalışması için hayati önemdedir. Şekil 2.1'de MEC aktarım mimarisi özetlenmiştir.

### **2.3 Kaynak Tahsisi**

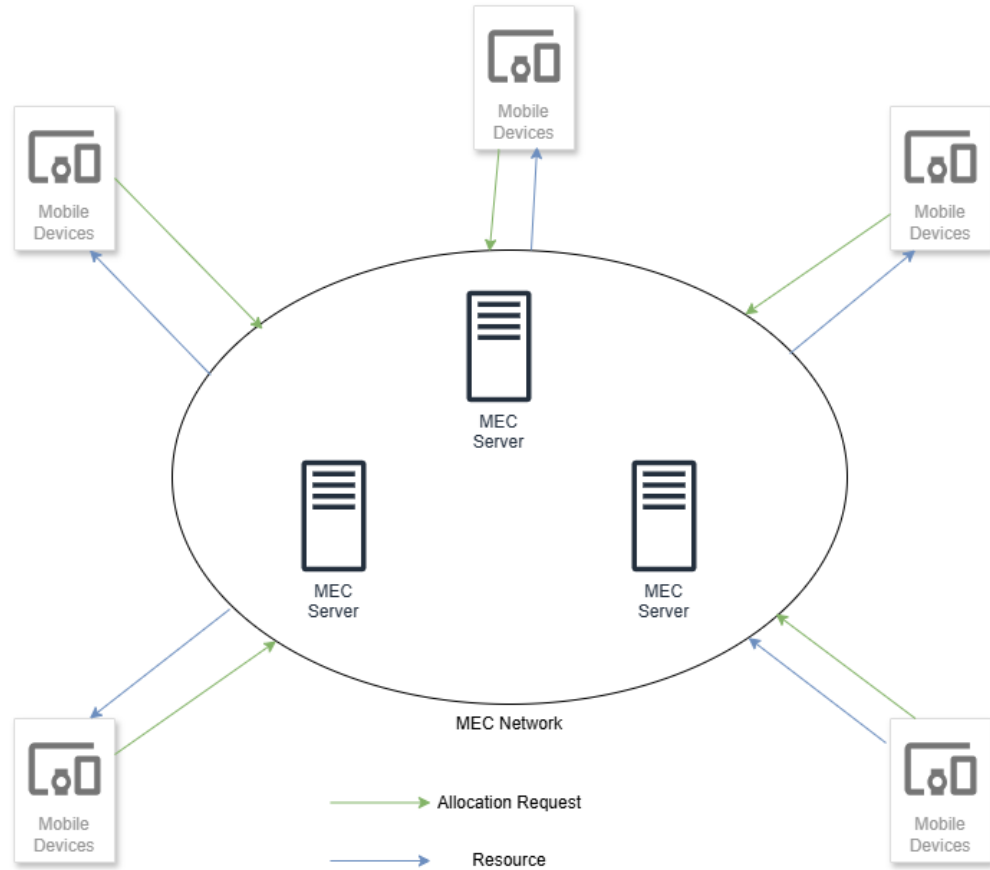
Kaynak tahsisi, Mobil Uç Bilişimin (MEC) kritik bir bileşenidir. Bu bileşen, kullanıcı taleplerini karşılamak için mevcut kaynakların ne ölçüde verimli kullanılacağını belirler. MEC bağlamında kaynaklar; hesaplama gücü, bellek, ağ bant genişliği ve depolama kapasitesini kapsar. Kaynak tahsisinin amacı, bu kaynakları sistem performansını optimize edecek, enerji tüketimini ve operasyonel maliyetleri en aza indirecek biçimde dağıtmaktır.

Araştırmacılar, kaynak tahsisi için genellikle dinamik optimizasyon yöntemlerine odaklanmıştır. Kaynak tahsisi araştırmalarında vurgulanan temel nokta, verimlilik açısından hesaplama aktarımı optimizasyonunu geliştirmektir. Bu durum, genellikle gecikme ölçütlerine odaklanan aktarım kararı çalışmalarından farklılık

göstermektedir. Kaynak tahsisinde temel endişe, genellikle enerji olan düğüm cihazlarının kaynak tüketimidir. İstemci cihazların çoğu pille çalışmaktadır.

Kaynak tahsisi ve aktarım kararı çalışmaları sıklıkla örtüşse de geliştirme hedefleri farklıdır. İlkinde odak noktası düğüm kaynak tüketimindeyken, ikincisinde gecikmedir. Bu ayrım, problem tanımında bir kaymaya yol açarak, belirli problem tanımına bağlı olarak Konveks veya Lyapunov optimizasyon yöntemlerini daha uygun çözümler haline getirmektedir (Al-Shuwaili vd. 2017) (Guo vd. 2017) (Liu vd. 2017) (Mao vd. 2017) (Wang vd. 2017) (Wu vd. 2019) (Zhao vd. 2017) (Zhou vd. 2019). Dinamik optimizasyon yöntemleri kaynak tahsisinde yaygın olmakla birlikte, öğrenme tabanlı çözümler de önemli bir yer tutmaktadır (Guo vd. 2018) (Pham vd. 2018) (Sun vd. 2018) (Tran vd. 2018) (Zhang vd. 2018) (Abdullaev vd. 2023).

Kaynak tahsisi, MEC ile kesişen klasik bir Nesnelerin İnterneti (IoT) problemidir. MEC mimarisinin mobilite yönü, bu konunun önemini diğer bilgisayar ağı dallarının ötesine taşımaktadır. Sonuç olarak, kaynak tahsisi MEC'in kritik bir yönüdür ve etkin yönetimi, mimarinin başarısı için hayati öneme sahiptir. Doğru kaynak tahsisi, gecikmeyi en aza indirmeye, ağ performansını optimize etmeye, enerji tüketimini azaltmaya ve kullanıcı deneyimini artırmaya yardımcı olabilir. Ancak, dinamik ve mobil bir ortamda kaynakları yönetmenin zorlukları, uyarlanabilir ve esnek kaynak tahsisi algoritmalarının geliştirilmesini gerektirmektedir. Şekil 2.2'de MEC mimarisi içinde kaynak tahsisini nasıl yapıldığı görselleştirilmiştir.



Şekil 2.2 Kaynak kullanımı

### 2.3.1 Kaynak tahsisi ile hesaplama aktarımı arasındaki ilişki

Kaynak tahsisi literatürü enerji ve kapasite kısıtlarına odaklanmaktadır (Al-Shuwaili vd. 2017; Mao vd. 2017; Liu vd. 2017). Hesaplama aktarımı ise gecikme önceliklidir (Mach vd. 2017; You vd. 2017). İki alan örtüşmektedir: sunucuya aktarım kararı, uç sunucunun anlık işlemci uygunluğu ( $cpu\_power\_availability$ ) ve bellek uygunluğu ( $memory\_availability$ ) değerlerine bağlıdır. Bu değerler karar değişkeni vektörü  $x$  içinde yer almaktadır (Bölüm 3.1).

Bu tezde kaynak tahsisi ayrı bir optimizasyon katmanı olarak çözülmez; kaynak kullanımı,  $F(x)$ 'in  $f_2$  bileşeni olarak modele dahil edilir. Böylece “düşük gecikme için aşırı kaynak harcama” ile “kaynak tasarrufu için yüksek gecikme” değiş tokuşu (trade-off) tek amaç fonksiyonunda birleştirilir (Bölüm 3.4). Düşük kaynak profillerinde

GA/ABC üstünlüğü, yüksek yoğunluk profillerinde PSO/GWO üstünlüğü bu birleşik ölçütün farklı ağırlıklandırıldığı durumlara karşılık gelmektedir (Bölüm 4.1.16).

## 2.4 Sunucu Konumlandırma

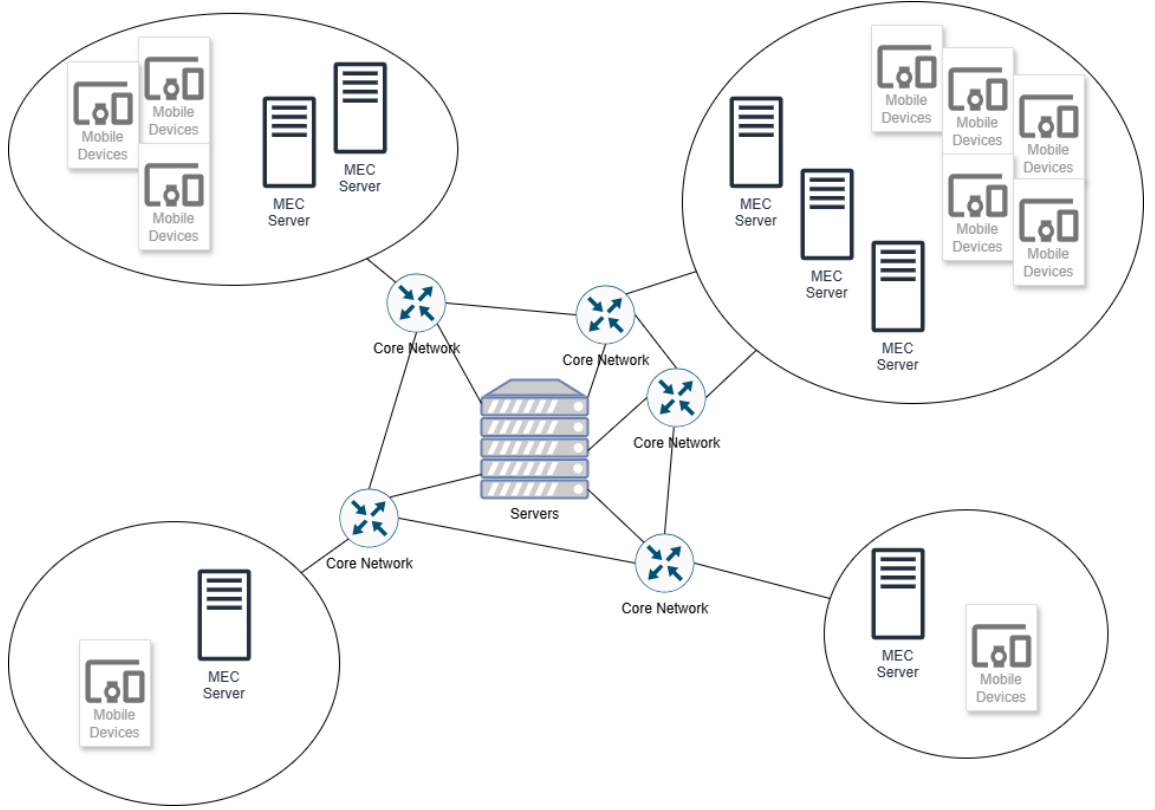
Sunucu konumlandırma, Mobil Uç Bilişim (MEC) mimarisinin önemli bir unsurudur. Uç cihazların hesaplama taleplerini etkili bir şekilde desteklemek için sunucuların stratejik olarak konumlandırılması çok önemlidir. Geleneksel bulut bilişiminin aksine, sunucular genellikle merkeziyken, MEC sunucularının uç cihazlara daha yakın konumlandırılması gerekmektedir, böylece gecikme azaltılır.

Sunucu konumlandırma, köklü bir ağ problemi olmasına rağmen, MEC bağlamında da özel bir öneme sahiptir. MEC bağlamında aktarım kararı veya kaynak tahsisi kadar yaygın bir şekilde incelenmemiş olsa da yine de araştırmacılar için ilgi konusu olmuştur. Uç sunucuların yerleştirilmesi, genel sistem performansı üzerinde doğrudan bir etkiye sahiptir; bu, hücresele ağlardaki baz istasyonlarının rolüne benzer ve sunucu dağıtımının kritik doğasını vurgulamaktadır.

MEC'in diğer zorluklarına kıyasla daha geleneksel doğası göz önüne alındığında, sunucu konumlandırma için çözümler genellikle geleneksel ağ yaklaşımlarına benzerlik göstermektedir. Bunlar arasında grafik tabanlı yöntemler (Hao vd. 2018), konfigürasyon optimizasyonu (He vd. 2021) (Deng vd. 2020) (Xiang vd. 2019), algoritmik çözümler (Läheranta vd. 2021) (Li vd. 2021) (Liu vd. 2020) (Wang vd. 2020) ve bant genişliği optimizasyonu (Li vd. 2020) yer almaktadır. Makine Öğrenimi (ML) teknikleri (Hao vd. 2018) (Kasi vd. 2021) (Deng vd. 2020) (Shao vd. 2020) ve dinamik optimizasyon yöntemleri (Zhou vd. 2019) de son çalışmalarla sunucu yerleştirmeye uygulanmıştır.

Sunucu konumlandırma, doğrudan aktarım sürecini etkilemese de genel sistem performansını artırmak, dolaylı olarak ve olumlu bir şekilde aktarımı etkileyebilir. Özetle, sunucu dağıtımı MEC mimarisinin önemli bir yönüdür ve sistem performansını ve verimliliğini optimize etmek için yeni stratejiler ve yaklaşımlar geliştirmeye yönelik devam eden araştırmalar vardır. Şekil 2.3 sunucu

konumlandırma işlemini açıklamaktadır.



Şekil 2.3 Sunucu konumlandırma

#### 2.4.1 Sunucu konumlandırmanın aktarım kararına dolaylı etkisi

Sunucu yerleşimi, istemci–sunucu mesafesini ve erişilebilir kapasiteyi belirlemektedir (Li vd. 2020; Lähderanta vd. 2021; Hao vd. 2018). Bu tezde distance ve server\_type alanları, konumlandırma sonucunun özet parametreleri olarak modellenmektedir. Ayrı bir yerleşim optimizasyonu çalıştırılmamaktadır. Dolayısıyla sunucu konumlandırma çalışmaları (Çizelge 2.3) bu tez için arka plan oluşturmaktadır. Asıl karar değişkeni x'in kendisidir.

Pratik yorum: Felaket Kurtarma ve Uzak Edge senaryolarında mesafe ve dinamik profil birlikte değişmektedir; GA'nın adaptasyon üstünlüğü (Bölüm 4.1) kısmen bu birleşik stres altında gözlemlenmektedir (Goldberg 1989). Fiber/5G kırsal senaryosunda ise altyapı yatırımı algoritma farklarını baskılamaktadır (Liu vd. 2020) (Taleb vd. 2017).

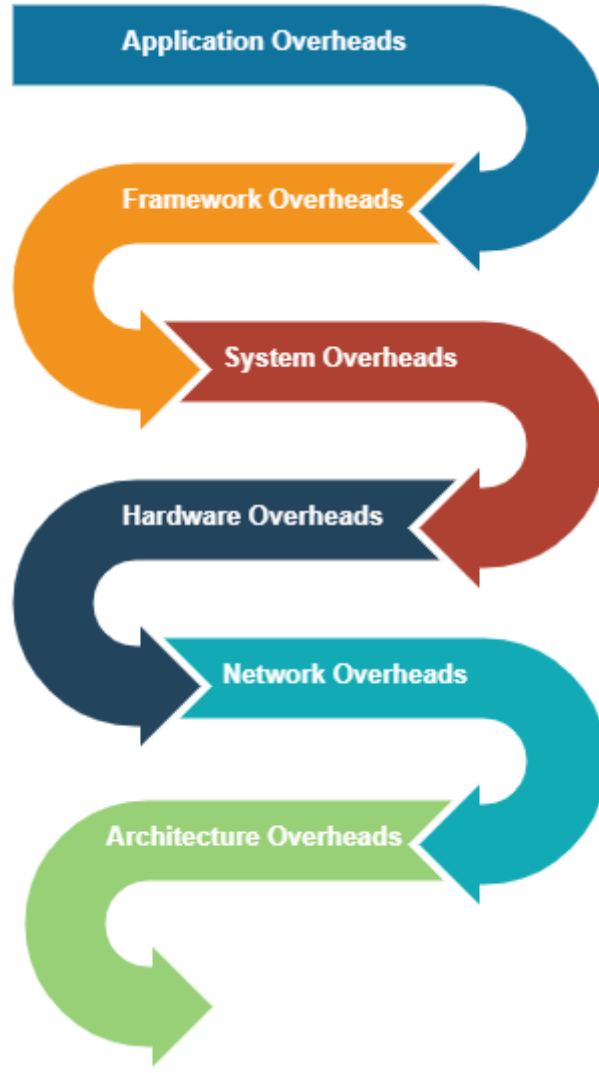
## 2.5 Ek Yük Yönetimi

Ek yük yönetimi, Mobil Uç Bilişim (MEC) mimarisinin kritik bir bileşenidir. MEC sistemleri, istenen performans seviyelerini koruyabilmek için verimli bir şekilde yönetilmesi gereken ek işlem ve iletişim aşırı yükleri getirir. Bu aşırı yükler, ek ağ atlamaları, artan işlem talepleri, ekstra depolama gereksinimleri ve uç düğümlerin güç tüketimi gibi faktörlerden kaynaklanan iletişim, hesaplama, depolama ve enerji aşırı yükleri gibi birkaç kategoriye ayrılabilir.

Önemine rağmen, ek yük yönetimi, MEC bağlamında diğer problemlere kıyasla nispeten daha az ilgi görmüştür. Bilgisayar bilimlerinde, bir sistemin karmaşıklığının, çevresel işlemlerin kaynak tüketimi ile doğru orantılı olduğu kabul edilen yerleşik bir problemidir. Aşırı yükü azaltmak, özellikle ağ alanında, sistemin veri iletim hızını doğrudan etkilemektedir. MEC'in bilgisayar ağlarının bir alt alanı olduğu göz önüne alındığında, ek yük yönetimi önemli bir çalışma alanı haline gelmektedir.

Araştırmacılar, güvenlik protokollerinin (Mora-Gimeno vd. 2018) optimizasyonu, veri tahsisi (Kavitha vd. 2020) (Hosseinzadeh vd. 2021) ve MEC ağ yapılandırması (Rimal vd. 2017) gibi çeşitli yaklaşımları incelemektedir. Makine Öğrenimi (ML) yöntemleri de ek yük yönetimi araştırmalarında ilgi görmeye başlamıştır (Yang vd. 2018) (Wang vd. 2019c). Ek yük yönetimi, doğrudan aktarım sürecini iyileştirmese de genel sistem performansını artırmak için kritik öneme sahiptir.

Sonuç olarak, etkili ek yük yönetimi, MEC sistemlerinin verimli bir şekilde çalışması için esastır. Aşırı yükler etkili bir şekilde yönetilerek, MEC sistemleri istenen performans seviyelerini elde edebilir, kullanıcı beklentilerini karşılayabilir, enerji tüketimini minimize edebilir ve hizmet kalitesini sürdürebilir. Ek yük yönetimi bünyesindeki çeşitli ek yükler Şekil 2.4'te görsel olarak paylaşılmıştır.



Şekil 2.4 Ek yük yönetimi

### 2.5.1 Ek yükün $F(x)$ modeline yansımaları

Ek yük yönetimi, iletişim ve işlem maliyetlerinin fazlasını kapsamaktadır (Mora-Gimeno vd. 2018) (Kavitha vd. 2020) (Rimal vd. 2017). Simülatörde “network\_queue”, “connection\_reliability” ve “crowd\_usage” gibi alanlar, paket gecikmesi ve yeniden iletim baskısını temsil etmektedir (Bölüm 3.4). Ek yük doğrudan optimize edilmemektedir;  $F(x)$  hesaplanırken otomatik yansımaktadır.

Bu yaklaşım, ek yükü ayrı bir RL problemi olarak çözmek yerine (Wang vd. 2019c), aktarım kararı ile değerlendirmeyi sağlamaktadır. Sonuç olarak yüksek kalabalık veya

düşük güvenilirlik profillerinde (Etkinlik Trafik, Sınırlı İnternet Kırsal) algoritma sıralaması değişmektedir; bu gözlem AS1 ile uyumludur (Bölüm 4.1).

## 2.6 Literatür Taraması Özeti

Hesaplama aktarımı, Mobil Uç Bilişim (MEC) bağlamında, diğer sorunlara göre bilgisayar biliminde nispeten daha yeni bir zorluk türüdür. Kaynak tahsisi ise bilgisayar biliminin en eski problemlerinden biridir ve modern işletim sistemlerinin ilk zamanlarına kadar uzanır. Benzer şekilde, sunucu yerleştirme de ilk ağ sistemlerinin başlangıcına kadar izlenebilir. Ek yük yönetimi, karmaşık sistemlerin ortaya çıkışından bu yana araştırmacıların ele aldığı bir kavram olup, uzun bir geçmişe sahiptir. İyi tanımlanmış olmalarına rağmen, bu zorluklar araştırmacılar için hala çok önemlidir. Tanımlanmış olmaları, onların önemini azaltmaz çünkü bu alanlardaki gelişmeler, sistem verimliliğinde önemli iyileştirmelere yol açabilir. Bu gerçek, onların önemini ve alaka düzeyini vurgulamaktadır. MEC bağlamında, bu alanlar araştırma için umut verici konular olmaya devam etmektedir. Ancak bu zorluklara kıyasla, hesaplama aktarımı hâlâ nispeten keşfedilmemiş bir araştırma alanıdır. Hesaplama aktarımı, MEC’de gecikmeyi azaltmadaki doğrudan etkisi nedeniyle ele alınması gereken kritik bir konudur. Bu sebeple bu tez çalışmasında hedef zorluk olarak seçilmiştir. Aşağıdaki çizelgeler dört ana MEC zorluğuna göre gruplanmıştır. Her çizelgede yöntemler ve ilişkili makaleler gösterilmektedir.

Hesaplama aktarımında veri yetersizliği nedeniyle ağırlıklı olarak takviyeli öğrenme yöntemi tercih edilmiştir. Bu zorluğun tabiatı gereği geleneksel optimizasyon metotları etkili olmadığı için diğer tercih edilen yöntemler de farklı öğrenme algoritmaları olmuştur. Detaylar Çizelge 2.1’de verilmiştir.

Çizelge 2.1 Hesaplama aktarım makaleleri

| Yöntem/ Yaklaşım                                                       | Makale                                                                                                                                                                                                                |
|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Takviyeli Öğrenme (Reinforcement Learning)                             | Zhang vd. 2019, Wang vd. 2019a, Zhao vd. 2020, Chen vd. 2019, Huang vd. 2018, Zhang vd. 2018, Vimal vd. 2020, Xiao vd. 2018, Feng vd. 2020, Cheng vd. 2019, Alam vd. 2019, Gao vd. 2024, Chen vd. 2023, Liao vd. 2023 |
| Matching/ Game/ Caching Theory                                         | Liao vd. 2019, Hao vd. 2018                                                                                                                                                                                           |
| Supervised/ Unsupervised/ Deep Learning, Swarm/ Heuristic Intelligence | Ale vd. 2019, Dong vd. 2019, Vimal vd. 2020, Wang vd. 2019b                                                                                                                                                           |

Kaynak tahsisi yalnızca MEC’e özgü bir konu olmadığı için hesaplama aktarımına göre daha geniş bir yöntem yelpazesinde araştırmalara konu olmuştur. Dinamik ve statik optimizasyon yöntemlerinin yanı sıra öğrenme algoritmaları da araştırmacılar tarafından kullanılmıştır. Yöntemler ve çalışmalar Çizelge 2.2’de listelenmiştir.

Çizelge 2.2 Kaynak tahsisi makaleleri

| Yöntem/ Yaklaşım                                                       | Makale                                                                        |
|------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Supervised/ Unsupervised/ Deep Learning, Swarm/ Heuristic Intelligence | Tran vd 2018, Shao vd. 2020                                                   |
| Convex/ Concave Optimization                                           | Al-Shuwaili vd. 2017, Guo vd. 2017, Wang vd. 2017, Wu vd. 2019, Zhao vd. 2017 |
| Lyapunov Optimization                                                  | Liu vd. 2017, Mao vd. 2017, You vd. 2017                                      |
| Matching/ Game/ Caching Theory                                         | Guo vd. 2018, Pham vd. 2018, Sun vd. 2018, Zhang vd. 2018                     |
| Software/ Configuration/ Architectural/ Network Enhancement            | Wu vd. 2019, Shojafar vd. 2016                                                |

Sunucu konumlandırma oldukça temel bir problem olduğu için araştırmacılar daha geleneksel yöntemleri tercih etmektedirler. Bunun yanı sıra çalışmaya uygun veri eksikliğinin de etkisi bu zorlukta da görülmüştür. Çizelge 2.3’te gösterildiği üzere araştırmacıların öğrenme yöntemi olarak RL/DRL tercih ettikleri gözlemlenmiştir.

Çizelge 2.3 Sunucu konumlandırma makaleleri

| Yöntem/ Yaklaşım                                            | Makale                                                                                                  |
|-------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| Takviyeli Öğrenme (Reinforcement Learning)                  | Kasi vd. 2021                                                                                           |
| Software/ Configuration/ Architectural/ Network Enhancement | Hao vd. 2018, Deng vd 2020, Xiang vd. 2019, Lähderanta vd. 2021, Li vd. 2021, Liu vd. 2020, Li vd. 2020 |

Ek yük yönetimi de sunucu konumlandırma ile benzer karakteristikler göstermektedir. Literatür taraması özeti Çizelge 2.4'te paylaşılmıştır.

Çizelge 2.4 Ek yük yönetimi makaleleri

| Yöntem/ Yaklaşım                                            | Makale                                                                                       |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| Takviyeli Öğrenme (Reinforcement Learning)                  | Wang vd. 2019c                                                                               |
| Software/ Configuration/ Architectural/ Network Enhancement | Mora-Gimeno vd. 2018, Kavitha vd. 2020, Hosseinzadeh vd. 2021, Rimal vd. 2017, Yang vd. 2018 |

### 2.6.1 Dört zorluk tablosunun sentezi ve tezdeki konumu

Çizelge 2.1–2.4 birlikte okunduğunda şu örüntü çıkar: (1) Hesaplama aktarımında RL/DRL ve derin öğrenme baskın; sezgisel/sürü zekâsı (Ale vd. 2019) (Dong vd. 2019) (Vimal vd. 2020) ikincil ama mevcuttur. (2) Kaynak tahsisinde konveks/Lyapunov ve oyun teorisi yaygındır (Guo vd. 2017) (Mao vd. 2017) (Pham vd. 2018). (3) Sunucu konumlandırma ve ek yükte mimari/yapılandırma ve RL karışıktır (Hao vd. 2018) (Deng vd. 2020) (Kasi vd. 2021). Bu tez, Çizelge 2.1'deki “Swarm/Heuristic Intelligence” (Sürü/Sezgisel Zeka) satırını genişleterek beş algoritmayı aynı protokolda toplamaktadır. Veri tarafında Shakarami vd. (2020) ve Mohammadi vd. (2018) ile uyumlu biçimde, eksik etiketli veri setleri standart öznitelik şemasına projekte edilmektedir (Bölüm 2.6.2; Bölüm 3.5.3). Carvalho vd. (2020) yapay zekâ tekniklerinin uç ortamlarında kullanımını vurgulamaktadır; bu çalışma sezgisel alt kümesine odaklanmış özel bir örnektir.

Dolayısıyla literatür özeti, yöntem seçimini “RL'ye alternatif değil, RL'nin yaygın olduğu alanda yorumlanabilir ve tekrarlanabilir karşılaştırma” olarak konumlandırır. Bu

konumlandırma, Bölüm 2.6.4 ve Bölüm 3'teki deneysel tasarımla tutarlıdır.

## 2.6.2 Gerçek dünya veri setleri ve MEC literatürü

Mobil uç bilişimde hesaplama aktarımı literatüründe “veri seti” terimi çoğu zaman ham veri kaydı (çalışma izi), ölçüm tablosu veya küme günlüğü anlamında kullanılmaktadır. Bu kaynaklar, doğrudan “yerelde mi, sunucuda mı yürütülsün” karar etiketini ve otuz iki boyutlu öznitelik vektörünü birlikte sağlamamaktadır (Mohammadi vd. 2018). Etiket eksikliği, takviyeli öğrenmenin (RL/DRL) yaygın biçimde tercih edilmesinin temel nedenlerinden biridir. Çünkü etiketli karar dizileri sentetik veya simülatör tabanlı ortamlarda üretilebilmektedir (Zhang vd. 2019; Wang vd. 2019a). Sezgisel ve evrimsel yöntemlerde ise literatürde yaygın yaklaşım, veri kaydından problem örneği (instance) türetmek ve tüm yöntemleri aynı amaç fonksiyonu  $F(x)$  altında karşılaştırmaktır (Ale vd. 2019; Dong vd. 2019).

Alibaba Cluster Trace Program (Guo vd. 2019), veri merkezi iş yükü CPU, bellek ve görev sürelerini içermektedir. Tipik kullanım alanları zaman çizelgeleme, kapasite planlama ve kaynak verimliliği analizidir. Azure Public Dataset — VM packing trace (Hadary vd. 2020), sanal makine yerleştirme ve paketleme kararlarını kaydeder. MEC bağlamında uç sunucu kapasite kısıtına indirgenebilir parametreler sağlamaktadır. GeoLife (Zheng vd. 2010), Microsoft araştırma GPS mobilite verisidir. Mesafe, hız ve kullanıcı hareket örüntüsü üretmektedir. Doğrudan görev boyutu vermemektedir, ancak coğrafi ve ağ değişkenleri için kanıt tabanı oluşturmaktadır. INFOCOM'23 5G ölçüm veri kümesi (Liu vd. 2023), saha grid ölçümleriyle kanal ve gecikme koşullarını temsil etmektedir. MLPerf Inference ve CNN/DailyMail türevli uç çıkarım yükleri (Hermann vd. 2015; Reddi vd. 2020), gecikme hassas yapay zeka görevlerinin uç sunucuya aktarım senaryolarına parametre sağlamaktadır.

Bu tezde ortak ilke şöyledir: ham veri seti kaydı  $\rightarrow$  standart otuz iki sütunlu öznitelik vektörü (STANDARD\_FEATURE\_COLUMNS)  $\rightarrow$  Python–Java simülatör  $\rightarrow$   $F(x)$ . Ham sonuç satırı üzerinde doğrudan popülasyon araması yapılmaz. Optimizasyon nesnesi Task ve simülatör skorudur. Bu, “veri setiyle optimizasyon” ifadesinin literatürdeki veri

seti koşullu örnek üretimi (trace-conditioned instance generation) anlamıyla uyumludur (Guo vd. 2019; Hadary vd. 2020). GSM operatör konum verilerinin yalnızca coğrafi boyut sağlaması, Bölüm 3.1'de açıklanan üretici zorunluluğunu tamamlayıcı biçimde desteklemektedir. Veri setleri eksik boyutları kontrollü doldurmaktadır. Üretici ise etiket ve karşılaştırma parametrelerini sağlamayı hedeflemektedir.

Çizelge 2.5'te özetlenen kaynak–MEC eşlemesi, şeffaflık ilkesine hizmet eder. Her sütun veya türetilmiş özellik, simülatördeki hangi alana (mesafe, task\_size, connection\_speed vb.) karşılık geldiği kodda ve Bölüm 3.5.3'te açıklanır; bu, karşılaştırılabilirlik iddiasının temelidir (Mao vd. 2017).

Çizelge 2.5 Gerçek dünya veri setleri ve tezdeki kullanım amacı

| Kaynak                 | Etiket       | Veriden çıkarılan özellikler      | MEC alt problemi       | Atıf                                 |
|------------------------|--------------|-----------------------------------|------------------------|--------------------------------------|
| Alibaba cluster v2018  | ALIBABA_V2   | CPU, bellek, görev süresi         | Kaynak kısıtlı aktarım | Guo vd. (2019)                       |
| Azure VM packing       | AZURE_PUBLIC | Yerleştirme, kapasite             | Uç kapasite / kuyruk   | Hadary vd. (2020)                    |
| GeoLife GPS            | CRAWDAD      | Mesafe, hız, yön                  | Mobilite, bağlantı     | Zheng vd. (2010)                     |
| INFOCOM'23 5G grid     | 5G_MONARCH   | Kanal, gecikme koşulu             | Ağ gecikmesi           | Liu vd. (2023)                       |
| CNN-DailyMail / MLPerf | MLPERF_EDGE  | Çıkarım yükü, gecikme hassasiyeti | İş yükü boyutu         | Hermann vd. (2015); Reddi vd. (2020) |

Sonuç olarak gerçek dünya veri setleri, Poisson senaryo üreticinin yerine geçmemektedir; onu tamamlamaktadır ve dış geçerlilik (external validity) için ikinci bir deneysel eksen oluşturmaktadır (Mach vd. 2017) (Taleb vd. 2017).

### 2.6.3 EdgeCloudSim, iFogSim ve özel simülatör tercihi

Uç bilişim, sis bilişim ve bulut bilişim performans değerlendirmesi için olgun simülatör araçları mevcuttur. EdgeCloudSim (Sonmez vd. 2017), IoT–sis bilişim–bulut bilişim

katmanlarında olay güdümlü simülasyon sunmaktadır. Karar değişkenleri çoğunlukla servis yerleşimi, yönlendirme ve uç sunucu dağıtımıdır. iFogSim (Gupta vd. 2017), IoT, sis bilişim ve bulut bilişim kaynak yönetimini modellemektedir. VM ve uygulama yerleşimi, enerji ve gecikme metrikleri üretmektedir. Dolui vd. 2017, sis bilişim, cloudlet ve MEC uygulamalarını karşılaştırarak her aracın farklı problem formülasyonuna hizmet ettiğini vurgulamıştır.

Bu tezde problem, tek bir görevin otuz iki öznitelik vektörü  $x$  ile kodlanır. Amaç fonksiyonu Bölüm 3.4'te verilen gecikme ve kaynak alt formüllerinin ağırlıklı birleşimidir. Her aday çözüm için yalnızca istemci ve uç sunucu yürütme yolları ayrı ayrı simüle edilmektedir. Daha düşük  $F(x)$  veren yol seçilir. Poisson tabanlı on beş senaryo üretici, HTTP üzerinden Python istemci – Java sunucu haberleşmesi ve veriden Görev projeksiyonu (Bölüm 3.5.3) EdgeCloudSim veya iFogSim'in hazır modüllerinde bulunmamaktadır.

Dolayısıyla bu araçlar uyumsuz değildir; araştırma sorusu farklıdır. Yerleşim ve ağ topolojisi odaklı çalışmalar için EdgeCloudSim ve iFogSim uygundur (Sonmez vd. 2017) (Gupta vd. 2017). Tez kapsamı Bölüm 1.1.1'de tanımlanmıştır. Bu kapsamda özel  $F(x)$  için özelleştirilmiş simülatör tercih edilmiştir (Şekil 3.1). Bu seçim, tüm algoritmaların aynı değerlendirme fonksiyonunu paylaşmasını ve referans çizgisi ile adil kıyası sağlamayı hedeflemektedir.

#### **2.6.4 Literatürden yönteme geçiş: Boşluk ve tez konumlandırması**

Hesaplama aktarımı literatüründe RL/DRL ve derin öğrenme baskındır (Zhang vd. 2019) (Wang vd. 2019a). Sezgisel ve sürü zekâsı tabanlı yöntemler Çizelge 2.1'de yer almakla birlikte, aynı  $F(x)$ , aynı karar değişkeni vektörü  $x$  ve aynı yerel/uç simülasyonu altında beş algoritmanın (GA, PSO, GWO, ACO, ABC) on beş senaryoda sistematik kıyasını sunan çalışma sayısı sınırlıdır (Shakarami vd. 2020). Veri tarafında gerçek dünya veri setleri (Alibaba, Azure, GeoLife, 5G ölçümü, çıkarım yükü) yaygın kullanılmakla birlikte, bu veri setlerinin tek başına karar etiketleri ve otuz iki MEC alanını birlikte vermediği Bölüm 2.6.2 ve Ek C'de gösterilmiştir (Guo vd. 2019).

Bu boşluk, tezin yöntem seçimini üç bileşenle şekillendirmiştir: (1) Açık formüllü  $F(x)$  ve değerlendirme parametreleri altında GA, PSO, GWO, ACO ve ABC'nin sistematik karşılaştırılması — RL/DRL yerine yorumlanabilir ve tekrarlanabilir bir meta-sezgisel çerçeve. (2) eksik öznitelik boyutlarını kontrollü tamamlayan Poisson tabanlı senaryo üretici. (3) dış geçerlilik için gerçek dünya veri setlerinin standart öznitelik şemasına yansıtıldığı adaptör katmanı. Bölüm 3 bu üç bileşeni tek deney protokolünde birleştirir; böylece Bölüm 4'teki sonuçlar hem senaryo hem veri seti ekseninde aynı araştırma sorularına yanıt vermektedir.

### 3. MATERYAL VE YÖNTEM

Bu bölümde tez çalışmasında kullanılmış olan veri seti ve simülasyonlar hakkında detaylı bilgi verilmiştir.

#### 3.1 Deneyisel Araştırma Tasarımı, Geçerlilik ve Tutarlılık

Bu bölüm, Bölüm 1'deki araştırma sorularını test etmek üzere tasarlanmış deneyisel araştırma mimarisini açıklamaktadır. Mimarinin ana bileşenleri aşağıda listelenmektedir:

- Bağımsız değişkenler: senaryo profili (on beş Poisson senaryosu) ve/veya veri seti kaynağı (beş gerçek dünya veri seti, Bölüm 3.5.3).
- Bağımlı değişkenler: simülatörden hesaplanan yanıt süresi ve kaynak kullanımı (Bölüm 3.4.2).
- Kontrol: referans çizgisi politikaları (3.5.2).
- Sabitler:  $F(x)$  ağırlıkları, batch=500 (senaryo) veya veri seti (harici), popülasyon=100, iterasyon=100.

Deney içi tutarlılık, tüm algoritmaların aynı standart öznitelik şeması, aynı simülatör, aynı  $F(x)$  ve hiperparametre limitleri, aynı referans çizgisi politikaları (Bölüm 3.5.2) ve aynı arama çözüm uzayı ile sağlanmaktadır. Böylece gözlenen farklar esasen operatör farkına indirgenir. Dış geçerlilik, on beş Poisson senaryosu ile beş gerçek dünya veri seti profilinde tekrarlanan koşullar ve hibrit tamamlama (Bölüm 3.5.3) ile desteklenmektedir. Mutlak gecikme değerleri saha ölçümü değildir, görelî sıralama ve profil eğilimleri yorumlanmaktadır (Mach vd. 2017). Yapı geçerliliği, otuz iki değişkenin ve  $F(x)$  bileşenlerinin MEC aktarım literatüründeki boyutlarla (görev, kaynak, ağ, mobilite) uyumlu tanımlanmasıyla güçlendirilmektedir (Mao vd. 2017).

#### 3.2 Veri Oluşturucu

Mobil Uç Bilişim görece yeni bir alan olduğu için bu konuda isterlere tam uyan

kullanılabilir veri literatür taraması sonucunda bulunamamıştır (Mohammadi vd. 2018) (Shakarami vd. 2020). Araştırmacıların yoğun bir şekilde takviyeli öğrenme yöntemini tercih etmelerinin sebebi de budur (Zhang vd. 2019) (Wang vd. 2019a). Mevcut kaynakların çoğu GSM operatörlerinin konum verisine dayanmaktadır. Görev boyutu, kaynak kısıtı ve aktarım kararı gibi MEC’e özgü alanları ya hiç vermemekte ya da kısmi sağlamaktadır. Alibaba ve Azure gibi veri setleri iş yükü parametreleri sunsa da doğrudan “yerelde mi, uçta mı?” etiketi taşımamaktadır (Guo vd. 2019) (Hadary vd. 2020). Bu nedenle literatürde araştırmacılar çoğunlukla simülatör veya rastgele sayı üreticileriyle temsili veri oluşturmayı seçmişlerdir. RL/DRL çalışmalarında ise ortam içi öğrenmeyi tercih etmektedir. Bu tezde deney içi tutarlılık ve tekrarlanabilirlik için kontrollü Poisson tabanlı senaryo üretici geliştirilmiştir. Dış geçerlilik için aynı üretici, Alibaba, Azure, GeoLife, 5G-MoNArch ve MLPerf temsilleriyle adaptör hattına bağlanarak ikinci deneysel eksen oluşturulmuştur (Bölüm 3.5.3).

Tam etiketli “yerel mi uçta mı” kararı içeren hazır MEC veri seti literatürde nadirdir. (Shakarami vd. 2020) Bu nedenle kontrollü Poisson üretici geliştirilmiştir. Ayrıca bu üretici tek başına bırakılmamış. Alibaba, Azure, GeoLife, 5G-MoNArch ve MLPerf temsilleri adaptör ve kalibrasyon ile standart öznitelik şemasına yansıtılarak ikinci bir deneysel eksen oluşturulmuştur (Bölüm 3.5.3). GSM konum verilerinin yalnızca coğrafi boyut sağlaması, üreticinin görev ve kaynak boyutlarını doldurma zorunluluğunu sürdürmektedir. Veri setleri eksik alanları tamamlamaktadır, veri üretici ise senaryo çeşitliliğini ve tekrarlanabilir  $\lambda$  profillerini sağlamayı hedeflemektedir.

MEC sistemlerinin çalışma koşullarını ve operasyonel süreçlerini modellemek amacıyla altı ana başlık altında toplam otuz iki değişken belirlenmiştir. Söz konusu değişkenler aşağıda sunulmuştur.

### **Coğrafi Değişkenler (Geographic Attributes)**

- Mesafe (Distance): İstemci (client) ile sunucu (server) arasındaki başlangıç mesafesidir. Gecikme süresi (delay time) ve ağ performansı hesaplamalarını etkilemektedir.
- Arazi Tipi (Terrain Type): Arazi türü, sinyal kalitesi ve bağlantı güvenilirliği hesaplamalarını etkilemektedir.
- Arazi Çetinliği (Terrain Harshness): Arazinin zorluğu, sinyal zayıflaması ve bağlantı kalitesi hesaplamalarını etkilemektedir.
- Hava Durumu (Weather): Hava koşulları, sinyal kalitesi ve bağlantı güvenilirliği hesaplamalarını etkilemektedir.
- Kullanıcı Yoğunluğu (User Density): Kullanıcı yoğunluğu, ağ trafiği ve kaynak kullanımını etkilemektedir.

Coğrafi değişkenler, istemci ile sunucu arasındaki konumsal ilişkiyi ve çevresel koşulları simülátöre aktaran beş sayısal alandan oluşmaktadır. Bu alanların kod adları, veri tipleri ve simülátörde kabul edilen değer aralıkları Çizelge 3.1'de görselleştirilmiştir.

Çizelge 3.1 Coğrafi değişkenler

| Değişken (kod adı)                  | Tip        | Değer aralığı / alınabilecek değerler           |
|-------------------------------------|------------|-------------------------------------------------|
| Mesafe (distance)                   | Doğal sayı | 1–200 (km); Poisson( $\lambda$ ) ile örneklenir |
| Arazi tipi (terrain type)           | Doğal sayı | 1–10 (ölçekli kategori)                         |
| Arazi çetinliği (terrain harshness) | Doğal sayı | 1–10                                            |
| Hava durumu (weather)               | Doğal sayı | 1–10                                            |
| Kullanıcı yoğunluğu (user density)  | Doğal sayı | 1–1000                                          |

### **Görev Değişkenleri (Task Attributes)**

- Görev Tipi (Task Type): Görev türü, işlem süresi ve kaynak kullanım sınıflarını belirlemektedir.
- Görev Büyüklüğü (Task Size): Görev boyutu, işlem süresi ve kaynak tüketimi hesaplamalarını doğrudan etkilemektedir.

- Görev Karmaşıklığı (Task Complexity): Görev karmaşıklığı, işlem süresi ve CPU kullanımını hesaplamalarını etkilemektedir.
- Tekrarlama Sıklığı (Recurrence): Görevin tekrarlanma sayısı, toplam işlem süresi ve kaynak kullanımını hesaplamalarını etkilemektedir.
- Gecikme Toleransı (Delay Tolerance): Kabul edilebilir maksimum gecikme süresi, sunucudaki görev zamanlamasını (task scheduling) etkilemektedir.

Görev değişkenleri, iş yükünün türünü, boyutunu ve zaman kısıtlarını tanımlamaktadır. Simülâtörde kullanılan görev alanlarının tipleri ve değer uzayları Çizelge 3.2'de özetlenmiştir.

Çizelge 3.2 Görev değişkenleri

| Değişken (kod adı)                   | Tip          | Değer aralığı / alınabilecek değerler |
|--------------------------------------|--------------|---------------------------------------|
| Görev tipi (task_type)               | Ayrık (enum) | COMPUTATION, STORAGE, MIXED           |
| Görev büyüklüğü (task_size)          | Doğal sayı   | 10–5000                               |
| Görev karmaşıklığı (task_complexity) | Doğal sayı   | 1–10                                  |
| Tekrarlama sıklığı (recurrence)      | Doğal sayı   | 1–20                                  |
| Gecikme toleransı (delay_tolerance)  | Doğal sayı   | 5–200 (ms)                            |

### **Yerel Kaynak Değişkenleri (Local Resource Attributes)**

- İşlemci (CPU): Yerel işlemci gücü, işlem hızını ve görevin tamamlama süresini etkilemektedir.
- Bellek (Memory): Yerel bellek miktarı, aynı anda işlenebilecek görev sayısını etkilemektedir.
- Disk Alanı (HDD Space): Disk alanı, veri depolama kapasitesini ve okuma hızını etkilemektedir.
- Ağ Adaptörü (Network Adapter): Ağ bağdaştırıcısının hızı, veri aktarım hızı hesaplamalarını etkilemektedir.
- Veriyolu Hızı (Bus Speed): Veri yolu hızı, dahili bileşenler arasındaki veri aktarım hızını etkilemektedir. Yürütme simülasyonu hesaplamalarında kullanılmaktadır.

Yerel kaynak değişkenleri, istemci cihazının işlem, bellek, depolama ve iletişim kapasitesini yüzdesel veya normalize biçimde temsil etmektedir. İlgili alanların tip ve değer sınırları Çizelge 3.3'te tablolastırılmıştır.

Çizelge 3.3 Yerel kaynak değişkenleri

| Değişken (kod adı)                       | Tip        | Değer aralığı / alınabilecek değerler (%) |
|------------------------------------------|------------|-------------------------------------------|
| İşlemci gücü (cpu_power)                 | Doğal sayı | 10–100                                    |
| Bellek (memory)                          | Doğal sayı | 10–100                                    |
| Disk alanı (hdd_space)                   | Doğal sayı | 10–100                                    |
| Ağ adaptörü hızı (network_adapter_speed) | Doğal sayı | 10–100                                    |
| Veriyolu hızı (bus_speed)                | Doğal sayı | 10–100                                    |

#### **Uç Sunucu Değişkenleri (Edge Server Attributes)**

- Sunucu Tipi (Server Type): Sunucu türü, işleme stratejisini belirlemektedir.
- İşlemci Uygunluğu (CPU availability): Sunucunun ilgili görev için kullanılabilir işlemci gücünü göstermektedir. İşlem kapasitesini etkilemektedir.
- Bellek Uygunluğu (Memory availability): Sunucunun ilgili görev için kullanılabilir bellek miktarını göstermektedir. İşlem kapasitesini etkilemektedir.
- Disk Uygunluğu (HDD availability): Sunucunun ilgili görev için kullanılabilir disk alanını göstermektedir. Veri depolama ve okuma hızını etkilemektedir.
- Veriyolu Hızı (Bus Speed): Dahili bileşenler arasındaki veri aktarım hızını etkilemektedir. Yürütme simülasyon hesaplamalarında kullanılmaktadır.
- Ağ Adaptörü (Network Adapter): Ağ bağdaştırıcısının hızı, istemci-sunucu iletişim hızı hesaplamalarını etkilemektedir.
- Ağ Yoğunluğu (Network Queue): Paket işleme gecikmesi hesaplamalarını etkilemektedir.
- Hareketlilik (Mobility): Sunucunun hareketliliği, bağlantı kararlılığını göstermektedir. Veri aktarım hesaplamalarını etkilemektedir.
- Hız (Speed): Sunucu hızı, bağlantı kalitesi ve gecikme hesaplamalarını etkilemektedir.

Uç sunucu değişkenleri, yerleşim tipini (LOCAL/EDGE) ve sunucu tarafı kaynak kullanılabilirliğini modellemektedir. Bu grubun dokuz alanına ait tip ve değer aralıkları Çizelge 3.4'te görselleştirilmiştir.

Çizelge 3.4 Uç sunucu değişkenleri

| Değişken (kod adı)                                         | Tip          | Değer aralığı / alınabilecek değerler |
|------------------------------------------------------------|--------------|---------------------------------------|
| Sunucu tipi (server type)                                  | Ayrık (enum) | LOCAL, EDGE                           |
| İşlemci uygunluğu (cpu power availability)                 | Doğal sayı   | %10–100                               |
| Bellek uygunluğu (memory availability)                     | Doğal sayı   | %10–100                               |
| Disk uygunluğu (hdd space availability)                    | Doğal sayı   | %10–100                               |
| Veriyolu uygunluğu (bus speed availability)                | Doğal sayı   | %10–100                               |
| Ağ adaptörü uygunluğu (network adapter speed availability) | Doğal sayı   | %10–100                               |
| Ağ kuyruğu (network queue)                                 | Doğal sayı   | 1–50                                  |
| Hareketlilik (mobility)                                    | Doğal sayı   | 0–10                                  |
| Hız (speed)                                                | Doğal sayı   | 0–30 (m/s)                            |

#### Ağ Değişkenleri (Network Attributes)

- Bağlantı Tipi (Connection Type): Bağlantı türü, maksimum veri aktarım hızını belirlemektedir.
- Bağlantı Hızı (Connection Speed): Bağlantı hızı, veri aktarım süresini etkilemektedir.
- Bağlantı Kalitesi (Connection Reliability): Bağlantı güvenilirliği, paket kaybı ve yeniden iletim oranını göstermektedir. Veri aktarımı hesaplamalarını etkilemektedir.
- Bağlantı Noktası Sayısı (Node Count): Bağlantı Noktası sayısı, ağın karmaşıklığını ve gecikme hesaplamalarını etkilemektedir.

Ağ değişkenleri, istemci–sunucu iletişiminin tipini, hızını, güvenilirliğini ve topoloji karmaşıklığını sayısal olarak ifade etmektedir. Dört ağ alanının tip ve değer uzayı Çizelge 3.5'te sunulmuştur.

Çizelge 3.5 Ağ değişkenleri

| Değişken (kod adı)                                 | Tip        | Değer aralığı / alınabilecek değerler |
|----------------------------------------------------|------------|---------------------------------------|
| Bağlantı tipi<br>(connection_type)                 | Enum       | 1=4G, 2=5G, 3=Fiber (1–5)             |
| Bağlantı hızı<br>(connection_speed)                | Doğal sayı | 50–10000 (Mbps)                       |
| Bağlantı güvenilirliği<br>(connection_reliability) | Doğal sayı | %30–100                               |
| Düğüm sayısı (node count)                          | Doğal sayı | 1–20                                  |

### **Dış Çevre Değişkenleri (Outer Environment Attributes)**

- Kullanıcı Hızı (User Speed): Kullanıcının hareket hızı, bağlantı kararlılığını ve sinyal kalitesini etkilemektedir. Ağ aktarım süresi hesaplamalarında kullanılmaktadır.
- Kullanıcı İstikameti (User Direction): Kullanıcının istikameti, istemcinin yönelimi ve sinyal gücünü etkilemektedir. Ağ aktarım süresi hesaplamalarında kullanılmaktadır.
- Kitle Frekansı (Crowd Frequency): Kalabalık sıklığı, ağ trafiğini ve kaynak kullanımını etkilemektedir.
- Kitle Kullanım Oranı (Crowd Usage): Kalabalığın ağ kullanım seviyesi, ağ yükü ve performansını etkilemektedir.

Dış çevre değişkenleri, kullanıcı mobilitesi ve kalabalık kaynaklı ağ yükünü yansıtmaktadır. Bu dört alanın simülatördeki tip tanımları ve değer aralıkları Çizelge 3.6'da görselleştirilmiştir.

Çizelge 3.6 Dış çevre değişkenleri

| Değişken (kod adı)                       | Tip        | Değer aralığı / alınabilecek değerler |
|------------------------------------------|------------|---------------------------------------|
| Kullanıcı hızı (user_speed)              | Doğal sayı | 0–20 (m/s)                            |
| Kullanıcı istikameti<br>(user_direction) | Doğal sayı | 0–360 (°)                             |
| Kitle frekansı<br>(crowd_frequency)      | Doğal sayı | 0–100                                 |
| Kitle kullanım oranı<br>(crowd_usage)    | Doğal sayı | 0–100                                 |

### 3.2.1 Poisson( $\lambda$ ) dağılımı: Katsayılar, gerekçe ve on beş senaryo profili

Senaryo tabanlı görev üretiminde, Bölüm 3.1’de listelenen sayısal görev alanları Poisson dağılımından örneklenir. On beş test senaryosunun her biri için merkezi bir konfigürasyonda tanımlı değerleri (örneğin görev büyüklüğü, kullanıcı yoğunluğu, bağlantı hızı) ilgili alanın Poisson ortalaması  $\lambda$  olarak yorumlanır; üretici her yeni görevde bu  $\lambda$  değerlerinden bir örnek çeker. Böylece aynı senaryo altında üretilen görevler ortak bir yoğunluk profiline bağlı kalır, ancak görevler arası varyasyon korunur. Uygulama NumPy rastgele sayı üreticisine dayanır. Poisson tercihinin kuramsal gerekçeleri aşağıda özetlenmektedir.

Normal dağılım tercih edilmemesinin gerekçeleri şunlardır: (1) Görev sayısı, kuyruk uzunluğu ve varış oranı gibi nicelikler negatif olamaz; Poisson doğal sayı ve sıfır kuyruk modelleriyle uyumludur (Mao vd. 2017). (2) MEC ve kablosuz ağ literatüründe Poisson varış süreçleri yaygın kabul görür (Liu vd. 2017) (Zhao vd. 2017). (3) Tek parametre ( $\lambda$ ) ile on beş senaryo arasında yorumlanabilir sistematik kıyas yapılabilir. (4) Sağa çarpık dağılım, meta-sezgisel algoritmaların sağlamlık(robust) testi için zorlayıcı başlangıç popülasyonları oluşturmaktadır (Goldberg 1989) (Rao 2009). Senaryo adları ve temsili  $\lambda$  merkezleri Çizelge 3.7’de listelenmiştir.

Çizelge 3.7 Senaryo adları ve temsili  $\lambda$  merkezleri

| Senaryo                                                   | $\lambda$<br>distance | $\lambda$ user<br>density | $\lambda$ task<br>size | $\lambda$<br>connection<br>speed | Profil<br>notu                     |
|-----------------------------------------------------------|-----------------------|---------------------------|------------------------|----------------------------------|------------------------------------|
| TEMEL (BASIC)<br>(Baz Senaryo)                            | 100                   | 50                        | 500                    | 500                              | Statik<br>referans                 |
| HIGH_DENSITY_URBAN<br>(Yüksek Yoğunluklu<br>Kentsel Alan) | 50                    | 500                       | 1000                   | 1000                             | Yoğun,<br>kısa<br>mesafe           |
| SMART_CITY<br>(Akıllı Şehir Ağı)                          | 75                    | 300                       | 800                    | 800                              | IoT<br>yoğun;<br>kısmen<br>dinamik |
| BUSINESS_DISTRICT<br>(Tüketici Ağı Sınırlı İş<br>Bölgesi) | 60                    | 200                       | 600                    | 600                              | Kurumsal<br>yük                    |

Çizelge 3.7 Senaryo adları ve temsili  $\lambda$  merkezleri (devam)

| Senaryo                                             | $\lambda$<br>distance | $\lambda$ user<br>density | $\lambda$ task<br>size | $\lambda$<br>connection<br>speed | Profil<br>notu      |
|-----------------------------------------------------|-----------------------|---------------------------|------------------------|----------------------------------|---------------------|
| EVENT_DRIVEN<br>(Etkinlik Kaynaklı Yüksek Trafik)   | 40                    | 1000                      | 400                    | 400                              | Geçici yük; dinamik |
| LIMITED_INTERNET<br>(Sınırlı İnternet Kırsal Bölge) | 150                   | 20                        | 100                    | 100                              | Düşük kaynak        |
| AGRICULTURAL<br>(Tarımsal Ağ İhtiyaçları)           | 120                   | 30                        | 150                    | 150                              | Düşük bant          |
| FIBER_5G<br>(Fiber/5G Altyapılı Kırsal Bölge)       | 100                   | 50                        | 900                    | 900                              | Yüksek bant         |
| TELEMEDICINE<br>(Kırsal Teletıp ve E-Öğrenme)       | 80                    | 40                        | 800                    | 800                              | Gecikme hassas      |
| SUBURBAN_MIXED<br>(Banliyö Ofis-Konut Ağı)          | 90                    | 150                       | 700                    | 700                              | Karma yük           |
| WORK_FROM_HOME<br>(Evden Çalışma ve Aile Ağı)       | 85                    | 120                       | 600                    | 600                              | Çoklu uygulama      |
| DATA_CENTER<br>(Büyükşehir Veri Merkezi)            | 30                    | 2000                      | 1200                   | 1200                             | Kurumsal yoğun      |
| DISASTER_RECOVERY<br>(Felaket Kurtarma Testi)       | 100                   | 500                       | 500                    | 500                              | Dinamik yük         |
| EDGE_COMPUTING<br>(Uzak Bölgelerde Uç Bilişim Ağı)  | 70                    | 80                        | 400                    | 400                              | Uç aktarım          |
| TEMPORARY_NETWORK<br>(Geçici Ağ Kurulumları)        | 90                    | 100                       | 300                    | 300                              | Mobil altyapı       |

Her senaryonun sözel tanımı Bölüm 3.5'te detaylandırılmıştır;  $\lambda$  profilleri bu tanımlarla tutarlı seçilmiştir. Poisson kullanımı, olay sıklığı kontrollü değiştirilerek sistemin farklı yük altındaki davranışını gözlemlemeyi mümkün kılmaktadır. Bu yaklaşım hem simülasyon temsil gücünü artırmaktadır hem de algoritma genellenebilirliğini test etmektedir.

### 3.2.2 Senaryo üreticinin meta-sezgisel algoritmalara uygunluğu

Senaryo üretici, meta-sezgisel aramanın giriş uzayını tanımlamaktadır: her birey otuz iki boyutlu öznitelik vektörü  $x$ 'tir ve uygunluk değeri simülatörden dönen  $F(x)$  ile ölçülmektedir (Bölüm 3.4). Hesaplama aktarımı,  $F(x)$  değerini minimize etme problemi

olarak formüle edilmektedir. Senaryo adı yalnızca başlangıç popülasyonunun  $\lambda$  profilini belirlemektedir. Yapılan her istekte Çizelge 3.7'deki Poisson konfigürasyonuna göre yeni bir görev objesi üretmektedir (Bölüm 3.2.1). Böylece algoritmalar aynı amaç fonksiyonu altında farklı yük profillerinde sistematik biçimde test edilebilmektedir.

Senaryo üreticinin meta-sezgisel yöntemlerle uyumu üç nedene dayanmaktadır. Birincisi, her aday çözüm otuz iki alanlı sabit bir karar değişkeni vektörü  $x$  olarak kodlanmaktadır. Amaç fonksiyonu için türev hesabı gerekmemektedir (Rao 2009). İkincisi, her aday için yürütme yolu tercihlerinin sonuçları simülatörde hesaplanmakta. Daha düşük sonuç veren yol seçilmektedir (Bölüm 3.4.4) (Mao vd. 2017). Üçüncüsü, on beş senaryo aynı  $F(x)$  ve aynı arama ayarları altında yalnızca başlangıç görev dağılımını değiştirmektedir. Böylece farklı yük koşulları kontrollü biçimde test edilebilmektedir (Shakarami vd. 2020).

Beş algoritma aynı görev kodlamasını paylaşmaktadır. Fark yalnızca algoritmaların çalışma şeklinde oluşmaktadır. GA'da kromozom, karar değişkeni vektörü  $x$ 'i kodlar. Çaprazlama ve mutasyon öznitelik düzeyinde uygulanmakta, elitizm en iyi bireyleri korumaktadır (Goldberg 1989). PSO'da parçacık konumu görevi temsil etmekte. Kişisel ve küresel en iyi sonuç değerleri hız güncellemesini yönlendirmektedir (Kennedy vd. 1995). GWO'da alfa-beta-delta kurtları en iyi üç çözüm etrafında güncelleme yapmaktadır. Yüksek kaynak senaryolarında agresif keşfe yatkındır (Mirjalili vd. 2014). ACO, feromon matrisiyle öznitelik alt uzayında yön tercihlerini biriktirmektedir (Dorigo vd. 2006). ABC ise işçi-gözcü-izci rolleri ve limit parametresiyle düşük kaynak profillerinde dengeli arama sağlamaktadır (Karaboga vd. 2007).

GA, PSO, GWO, ACO ve ABC aynı  $F(x)$ , aynı değerlendirme protokolü ile aynı popülasyon (100) ve iterasyon (100) limitini paylaşmaktadır. Gözlemlenen performans farkları esasen operatör kaynaklı yorumlanmaktadır. Gerçek dünya veri seti deneylerinde veri setlerinden görev örnekleri üretilmekte. Bu akış “ScenarioConfig” döngüsünden ayrı tutulmaktadır (Bölüm 3.5.3). Her iki deneysel eksende de uygunluk hesabı aynı simülatör çekirdeği üzerinden yürütülmektedir.

### 3.2.3 Değişken grupları ve aktarım kararına etkisi

Otuz iki değişkeni altı başlıkta düşünmek, tez boyunca okumayı kolaylaştırmaktadır. Coğrafi ve dış çevre grubu (mesafe, arazi, hava, kullanıcı yoğunluğu, hız, yön, kalabalık) iletişim gecikmesini ve sinyal kalitesini etkilemektedir. Ağ grubu (bağlantı tipi, hız, güvenilirlik, düğüm sayısı) verinin ne kadar sürede taşınacağını belirlemektedir. Görev grubu (tip, boyut, karmaşıklık, tekrar, gecikme toleransı) işin ne kadar ağır olduğunu tanımlamaktadır. Yerel kaynak grubu (istemci CPU, bellek, disk, adaptör, veriyolu) “telefonda çalıştırırsam ne olur?” sorusunu; uç sunucu grubu (sunucu tipi, CPU/bellek/disk uygunluğu, kuyruk, hareketlilik) “uç sunucuya verirsem ne olur?” sorusunu besler (Mao vd. 2017) (Mach vd. 2017).

Meta-sezgisel algoritma bu gruplardaki sayıları birlikte değiştirir. Tek bir alanı (örneğin yalnızca mesafe) optimize etmez. Örneğin yüksek görev hacmi ve düşük bağlantı hızı bir araya geldiğinde EDGE maliyeti artabilir. Algoritma x'i öyle bir bölgeye çeker ki yerel yürütme daha düşük  $F(x)$  üretebilmekte veya tersi gerçekleşebilmektedir (You vd. 2017). Poisson üretici (Bölüm 3.1.1), bu grupların senaryo bazında tipik değer aralıklarını kontrollü üretmektedir. Böylece “kırsal düşük kaynak” ile “kentsel yüksek yoğunluk” aynı kodla test edilebilmektedir (Liu vd. 2020) (Taleb vd. 2017).

### 3.3 Simülasyon Düzenegi

Çalışmanın yapılabilmesi için veri oluşturucuya ek olarak bir simülasyon ortamı da hazırlanmıştır. İstemci sunucu olmak üzere iki ana bileşenden oluşmaktadır. Bunlardan istemci bileşeni MEC ortamındaki kullanıcıları temsil etmektedir ve karar mekanizmasının çalışmasını simüle etmekle yükümlüdür. Sunucu ise MEC ortamındaki uç sunucuları simüle etmektedir. Gelen görevlerin temsili olarak gerçekleştirilmesini sağlamaktadır.

İstemci bileşeni python programlama dilinde, sunucu bileşeni java programlama dilinde yazılmıştır. İstemci bileşenin karar mekanizmaları hariç python yerleşik

kütüphaneleri kullanılmıştır. Üç alt bileşenden oluşmaktadır.

- Veri Dağıtıcısı
- Controller
- Sonuç Analizcisi

Veri dağıtıcısı, isminden anlaşılacağı üzere veri oluşturucusunun yarattığı verilerle sistemi beslemekten sorumludur ve veri oluşturucusu ile entegre çalışmaktadır. Genel akış, kontrol alt bileşeni tarafından yönetilmektedir. Controller bileşeni tüm sistemin yönetildiği kısımdır. Veri oluşturucusundan gelen görevleri sunucuya ileterek sonuçlarını analiz bileşenine aktarır. İstemci ile sunucu arasındaki haberleşme HTTP kullanılarak simüle edilmiştir. Ancak istenirse farklı protokollerle de haberleşmeye açıktır. Ayrıca aktarım kararını vermek de bu bileşenin sorumluluğundadır. Son olarak Sonuç Analizcisi, yine isminden de anlaşılacağı gibi sonuçları analiz ederek yorumlamaya uygun hale getirmektedir.

### 3.3.1 Simülasyon bileşenlerinin işlevsel ayrıntıları

Veri dağıtıcısı, “ScenarioConfig” veya harici “task\_factory” üzerinden “Controller'a” iletir. “Controller”, her aday çözüm için görevleri simülatöre gönderir. Dönen gecikme ve kaynak skorları amaç fonksiyonu ile birleştirilir (Bölüm 3.4). HTTP iletişimi, gerçek MEC API gecikmesini milimetrik modellemez; ancak tüm algoritmalar için aynı ağ katmanı maliyetini taşır (Mach vd. 2017).

Sunucu bileşeni (Java/Spring Boot), delay ve execution alt formüllerini uygular. İstemci tarafı optimizasyon döngüsünü yürütür. Sonuç analizcisi, koşu sonuçlarını yoruma uygun formatlara dönüştürür (Şekil 4.1–4.36). Gerçek dünya veri seti hattında da akış aynıdır. Yalnızca görev kaynağı değişmektedir (Bölüm 3.5.3). Bu mimari, EdgeCloudSim'deki olay kuyruğu yerine tek görev–tek skor döngüsüne optimize edilmiştir (Sonmez vd. 2017) (Bölüm 2.6.3).

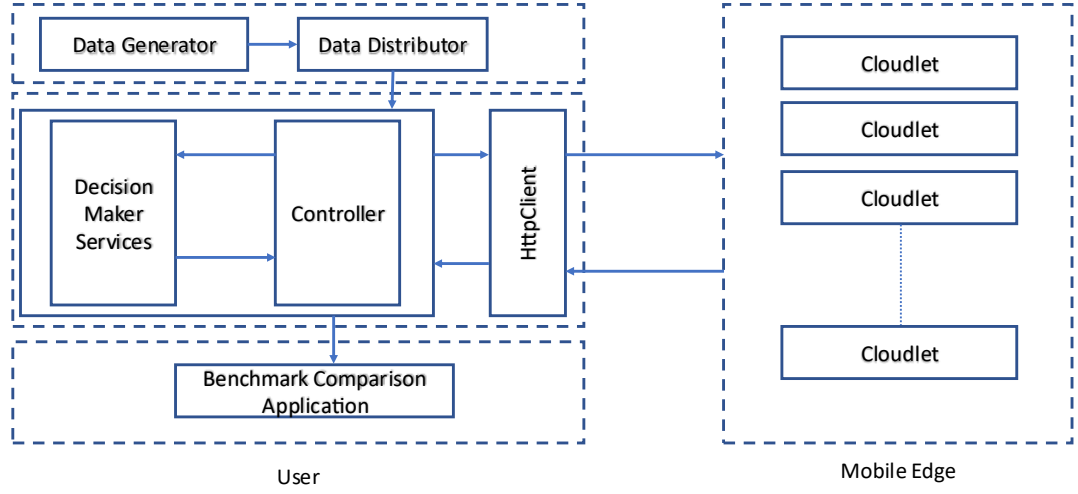
### 3.3.2 Bileşenlerin sorumlulukları

Veri dağıtıcısı, Bölüm 3.1'deki veri üreticiden görev örnekleri alır. Görev otuz iki değerden oluşan  $x$  vektörüdür. Controller, seçilen meta-sezgisel algoritmayı çalıştırır: popülasyondaki her  $x$ 'i Java sunucusuna HTTP ile gönderir, dönen gecikme ve kaynak skorlarını amaç fonksiyonuna aktarır. Daha sonra algoritmalar bir sonraki nesilleri üretirler (Şekil 3.1). Sunucu (Spring Boot), kendisine gelen görevlerden gecikme ve yürütme alt formüllerini uygular (Bölüm 3.4). Sonuç analizcisi, koşu sonuçlarını grafiklere ve tablolara dönüştürür (Bölüm 4.1).

Bu ayrım önemlidir: optimizasyon Python tarafında, fiziksel maliyet modeli Java tarafındadır. Tüm algoritmalar aynı sunucu formüllerini kullandığı için fark, arama stratejisinden kaynaklanır — bu da AS2'nin deneysel mantığıdır (Rao 2009) (Mao vd. 2017).

Sunucu bileşeni ise Spring Boot kullanılarak geliştirilmiştir. İstemciden gelen görev şablonlarını simülasyon formüllerine koşmakla yükümlüdür. Genel mimari tasarım aşağıda paylaşılmıştır (bkz. Şekil 3.1). Şekil 3.1'de sunulan simülasyon mimarisi, hesaplama aktarımı probleminin değerlendirilmesi için iki ana bileşenden oluşturulmuştur: MEC kullanıcılarını ve karar mekanizmasını temsil eden Python istemci ile uç sunucuları modelleyen Java tabanlı sunucu. Veri oluşturucudan elde edilen örnek vektörleri veri dağıtıcı aracılığıyla “Controller’a” iletilmekte; “Controller”, seçilen meta-sezgisel algoritma ile aday çözümleri üretmekte ve her aday için HTTP üzerinden sunucuya göndererek yerel ve uç yürütme yollarının maliyetini hesaplatmaktadır. Nihai sonuçlar sonuç analizcisi tarafından raporlanmaktadır.

## Sistem Tasarımı



Şekil 3.1 Simülasyon mimaris

### 3.4 Karar Algoritmaları

MEC ortamı değişkenliğin etkili olduğu bir ortam olduğu için karar mekanizması optimizasyonu için sezgisel algoritmalar (Heuristic Algorithms) tercih edilmiştir. Bu tez çalışması kapsamında aktarım kararını vermek için Genetik Algoritma (Genetic Algorithm), Parçacık Sürüsü Optimizasyonu (Particle Swarm Optimization), Gri Kurt Algoritması (Gray Wolf Algorithm), Karınca Kolonisi Optimizasyonu (Ant Colony Optimization) ve Yapay Arı Kolonisi (Artificial Bee Colony) kullanılmıştır. Sezgisel algoritmalarla ait hiperparametreler algoritmaların başarısını doğrudan etkilemektedir. Bu bağlamda hiperparametre değerlerine karar verilirken deneysel yöntemler (Grid/Random Search) uygulanmıştır. Elde edilen değerler literatürdeki değerler ile karşılaştırılarak uygunluğuna karar verilmiştir. Aşağıda her algoritma için detaylar paylaşılmıştır.

#### 3.4.1 Beş algoritma seçiminin gerekçesi ve ortak protokol

Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO), Gri Kurt Optimizasyonu (GWO), Karınca Kolonisi Optimizasyonu (ACO) ve Yapay Arı Kolonisi (ABC) algoritmaları, sırasıyla evrimsel, sürü zekası ve koloni tabanlı optimizasyon yaklaşımlarını temsil eden yaygın meta-sezgisel yöntemlerdir (Goldberg 1989; Kennedy

vd. 1995; Mirjalili vd. 2014; Dorigo vd. 2006; Karaboga vd. 2007). Bu algoritmaların tamamı, sürekli ve ayrık değişkenlerden oluşan karma yapıları karar değişkeni vektörleri  $x$  üzerinde arama gerçekleştirebilmekte ve çözüm kalitesini değerlendirmek amacıyla simülatör tarafından üretilen skaler amaç fonksiyonu değeri  $F(x)$ 'i kullanmaktadır (Rao 2009).

Algoritmaların adil ve karşılaştırılabilir biçimde değerlendirilebilmesi amacıyla tüm deneylerde ortak bir deneysel protokol uygulanmıştır. Bu kapsamda popülasyon büyüklüğü 100, maksimum iterasyon sayısı 100 ve senaryo büyüklüğü 500 olarak sabit tutulmuştur. Tüm yöntemler aynı gerçek dünya veri setleri üzerinde çalıştırılmış ve her algoritma için elde edilen en iyi çözüm raporlanmıştır (Bölüm 3.5.2). Böylece algoritmalar arasındaki performans farklılıklarının deneysel parametrelerden değil, yalnızca algoritmaların özgü arama mekanizmalarından kaynaklanması sağlanmıştır. Bu mekanizmalar; GA için elitizm ve genetik operatörler, PSO için hız ve konum güncelleme stratejileri, GWO için kurt hiyerarşisi temelli liderlik modeli, ACO için feromon tabanlı öğrenme yapısı ve ABC için limit değeri ile gözcü arı davranışlarıdır. Bu yaklaşım, araştırma sorusu AS2 kapsamında algoritmaların farklı koşullar altındaki göreceli performanslarının nesnel olarak değerlendirilmesine olanak sağlamaktadır.

### 3.4.2 Genetik algoritma (Genetic Algorithm)

Genetik Algoritma (GA) 1975 yılında J.H. Holland tarafından geliştirilmiştir (Holland 1975). GA bir problemin olası çözüm uzaylarından oluşan bir başlangıç popülasyonu ile başlamaktadır. Doğal seçilime benzer bir yaklaşım uygular. Her birey problemin doğasına göre tasarlanmış bir uygunluk fonksiyonuna göre değerlendirilmektedir. Uygun bulunan adaylar bir sonraki nesil olarak seçilir. Sık kullanılan seçim yöntemleri arasında rulet tekerleği seçimi, turnuva seçimi ve sıralama seçimi bulunur. Bu aşamada, yüksek uygunluk değerine sahip bireylerin seçilme olasılığı daha yüksektir, bu da iyi çözümlerin bir sonraki nesle aktarılmasını sağlamaktadır. Çaprazlama ve mutasyon araçları ile probleme en uygun çözüme ulaşmaya çalışır.

Genetik Algoritmada 5 hiperparametre kullanılmaktadır. Bunlar; nüfus büyüklüğü

(population\_size), iterasyon sayısı (max\_iterations), elit oranı (elite\_size), mutasyon oranı (mutation\_rate) ve çaprazlama oranı(crossover\_rate). Yüksek popülasyon ve iterasyon, algoritmanın daha iyi sonuç üretmesini sağlamaktadır ancak hesaplama süresini artırmaktadır. Elitizm ve yüksek çaprazlama oranı hızlı yakınsama sağlamaktadır, mutasyon ise çeşitliliği korur. Hiperparametrelerin alacağı değerler algoritmanın başarısını doğrudan etkilemektedir. Goldberg yaptığı çalışmada GA için hiperparametre değer aralıklarını aşağıdaki gibi vermiştir. (Goldberg 1989)

- Nüfus Büyüklüğü: 20-100
- Çaprazlama Oranı: 0.6–0.9
- Mutasyon Oranı: 0.001–0.1
- Elit Oranı: %5–%20

Bu tez kapsamında parametre değerlerine karar verilirken deneysel yöntemlerden Grid Search kullanılmıştır. Elde edilen değerler aşağıda listelenmiştir:

- Nüfus Büyüklüğü: 100
- Çaprazlama Oranı: 0.8
- Mutasyon Oranı: 0.05
- Elit Oranı: %10

Elde edilen değerlerin literatür ile uyumlu olduğu görülmüştür. GA'ya ait sonuçlar Test Sonuçları bölümünde verilecektir.

### 3.4.3 Genetik algoritma (GA) — MEC aktarım probleminde uygulama adımları

Uygunluk değeri simülatörden dönen Denklem (3.23) ile hesaplanmaktadır.

Bu tezde GA, otuz iki boyutlu karar değişkeni vektörü  $x$  üzerinde çalışmaktadır.  $t$  neslinde  $i$ . bireyin uygunluğu  $F(x_i^t)$  ile ölçülmekte; küçük değer üstün kabul edilmektedir. Elitizm ile en iyi  $N_e$  birey Denklem (3.1) ile belirlenmekte ve doğrudan  $t+1$  nesline aktarılmaktadır ( $N=100$  için  $N_e=10$ ).

$$N_e = \lfloor 0,1 \cdot N \rfloor \quad (3.1)$$

Turnuva seçiminde  $k=3$  adaydan biri ebeveyn olarak seçilmektedir; %80 olasılıkla en düşük  $F(x)$  değerine sahip birey, %20 olasılıkla ikinci en iyi birey alınmaktadır.

Çaprazlama olasılığı  $p_c=0,8$  ile uygulanmaktadır.  $d$  özniteliği için uniform çaprazlama:

$$x_{child,d} = \begin{cases} x_{e1,d} & r < 0,5 \\ x_{e2,d} & r \geq 0,5 \end{cases} \quad (3.2)$$

Burada  $r \sim U(0,1)$  olup  $e1$  ve  $e2$  ebeveynlerdir. Sunucu tipi (yerel/uç) özniteliği ayrıca %50 olasılıkla ikinci ebeveyninden devralınmaktadır.

$$x'_d = x_d \cdot u, u \sim U(0,9; 1,1) \quad (3.3)$$

- 1) Başlangıç: generate\_task\_for\_scenario(s) veya task\_factory() ile 100 görev üretilir.
  - 2) Değerlendirme: simülatör  $F(x)$  hesaplar.
  - 3) Seçim:  $k=3$  turnuva seçimi ile ebeveyn belirlenmekte; elite\_size=10 en iyi birey korunmaktadır (Goldberg 1989).
  - 4) Çaprazlama: crossover\_rate=0,8 ile öznitelik karışımı.
  - 5) Mutasyon: mutation\_rate=0,05 ile çarpımsal uniform pertürbasyon uygulanmaktadır.
  - 6) Yineleme: max\_iterations=100 veya yakınsama.
  - 7) Çıktı: En düşük  $F(x)$  ve yerel/uç kararı.
- Literatür uyumu: Goldberg (1989); Holland (1975).

#### 3.4.4 Parçacık sürüsü optimizasyonu (Particle Swarm Optimization)

Parçacık Sürüsü Optimizasyonu (PSO), 1995 yılında James Kennedy ve Russell Eberhart tarafından geliştirilen bir evrimsel optimizasyon algoritmasıdır. Doğal yaşamda kuş sürülerinin ve balina gruplarının davranışlarını model olarak geliştirilmiştir. PSO, çok sayıda çözüm adayının (birey) birbiriyle etkileşimde bulunarak en uygun çözümü bulmaya çalıştığı bir arama algoritmasıdır. (Kennedy vd. 1995) PSO doğrusal olmayan optimizasyon problemlerine etkili çözüm üretmektedir. Başlangıç popülasyonundaki her birey kendisi için en iyi çözüme ulaşmaya çalışır. Bu çözümün yanında bireyler sürü

içindeki en iyi çözümle birlikte sürekli olarak hem kendi çözümlerini hem de sürü çözümlerini güncellerler. Süreç çözüm kalitesi istenen seviyeye kadar tekrar eder.

PSO 9 adet hiperparametreye sahiptir: Parçacık sayısı(population\_size), iterasyon sayısı(max\_iterations), atalet katsayısı aralığı ( $w_{max}$ ,  $w_{min}$ ), bilişsel katsayı aralığı ( $c1_{start}$ ,  $c1_{end}$ ), sosyal katsayı aralığı ( $c2_{start}$ ,  $c2_{end}$ ) ve maksimum hız( $v_{max}$ ). Atalet katsayısı başta keşif için yüksektir, sonraki iterasyonlarda yakınsama için düşürülür. Bilişsel ve sosyal katsayılar bireysel ve toplu öğrenmeyi dengeler. Hız sınırı ise her iterasyondaki büyük adımları engelleyerek dengeli öğrenmeyi sağlamaktadır. Literatüre göre önerilen hiperparametre değerleri aşağıda listelenmiştir (Kennedy vd. 1995) (Shi vd. 1998):

- Nüfus Büyüklüğü: 20-50
- Atalet Katsayısı: 0.9-0.4 (lineer azalan)
- Bilişsel Katsayı: 1.5–2.0
- Sosyal Katsayı: 1.5–2.0
- Maximum Hız: 20

Deneysel yöntemler ve literatür beraber değerlendirilerek PSO hiperparametrelerine karar verilmiştir. PSO özelinde sadece nüfus büyüklüğü tüm iyileştirici (optimizer)ler için ortak olması için 100 seçilmiştir. Bu durum sadece hesaplama süresini artırmıştır. PSO'ya ait sonuçları Test Sonuçları bölümünde verilecektir.

### **3.4.5 Parçacık sürüsü optimizasyonu (PSO) — MEC aktarım probleminde uygulama adımları**

PSO adımları, Bölüm 3.4'teki ortak  $F(x)$  çerçevesi altında yürütülür; parçacık konumu karar değişkeni vektörü  $x_i$ , hız güncellemesi ise arama yönünü belirlemektedir (Kennedy vd. 1995).

PSO uygulamasında her parçacık  $i$ , konum vektörü  $x_i$  ve hız vektörü  $v_i$  ile temsil edilmektedir. Sürekli 21 öznitelik boyutu için (coğrafi, görev, yerel, ağ ve dış ortam

grupları) güncelleme denklemleri aşağıda verilmiştir; sunucu tipi hız güncellemesine dahil edilmemekte, F(x) değerlendirmesi sırasında yerel ve uç yolları kıyaslanarak belirlenmektedir.

t iterasyonunda atalet ve öğrenme katsayıları lineer olarak güncellenmektedir:

$$w(t) = w_{\max} - (w_{\max} - w_{\min}) \cdot \frac{t}{T} \quad (3.4)$$

$$c_1(t) = c_{1,b} - (c_{1,b} - c_{1,s}) \cdot \frac{t}{T} \quad (3.5)$$

$$c_2(t) = c_{2,b} + (c_{2,s} - c_{2,b}) \cdot \frac{t}{T} \quad (3.6)$$

d boyutu için hız güncellemesi (uygulama kodu):

$$r_1 \sim U(0,1), r_2 = 1 - r_1 \quad (3.7)$$

$$q_d = 0,1 \cdot N(0; 0,5) \cdot |x_{gbest,d} - x_{i,d}^t| \quad (3.8)$$

$$v_{i,d}^{t+1} =$$

$$\text{clip}(w(t)v_{i,d}^t + c_1(t)r_1(x_{pbest,i,d} - x_{i,d}^t) + c_2(t)r_2(x_{gbest,d} - x_{i,d}^t) + q_d, -v_{\max}, v_{\max}) \quad (3.9)$$

$v_{\max} = 20$ . Konum güncellemesi:

$$x_{i,d}^{t+1} = x_{i,d}^t + v_{i,d}^{t+1} \quad (3.10)$$

ardından  $x_i$  simülatör sınır aralıklarına kırpılmaktadır. Kişisel en iyi (pbest) ve küresel en iyi (gbest), F(x) küçüldüğünde güncellenmektedir.

- 1) Parçacıklar görev konumları; hız vektörleri öznitelik boyutunda.
- 2) pbest ve gbest F(x) ile güncellenir (Kennedy vd. 1995).
- 3)  $w(t)$ ,  $c_1(t)$  ve  $c_2(t)$  lineer zamanla değişmekte; kuantum terimi ve tamamlayıcı öğrenme ( $r_2=1-r_1$ ) kod uygulamasına göre kullanılmaktadır (yukarıdaki denklemler).
- 4) Konumlar simülatör kabul aralığına kırpılır.
- 5) Yüksek yoğunluklu senaryolarda hızlı iyileşme; düşük kaynakta aşırı kaynak tüketimi riski.

Literatür uyumu: Kennedy vd. (1995); Shi vd. (1998).

### 3.4.6 Gri kurt algoritması (Gray Wolf Algorithm)

Gri Kurt Algoritması (GWO), 2014 yılında Seyedali Mirjalili ve ekibi tarafından geliştirilen doğadan ilham alınmış bir optimizasyon algoritmasıdır. GWO, gri kurtların avlanma davranışlarını taklit ederek optimum çözümler bulmayı hedefler. (Mirjalili vd. 2014) GWO, gri kurtların avlanma stratejilerini modelleyerek optimizasyon yapar. Gri kurtlar, sosyal hiyerarşi ve iş birliği içinde avlarını avlarlar. Bu davranış, algoritmanın temel çalışma prensiplerine ilham vermiştir. GWO'nun temel adımları şunlardır:

- Başlangıç Popülasyonu:
- Kurt Hiyerarşisi:
- Av Konumunun Güncellenmesi:
- Dairesel Hareket ve Saldırı:
- Sonuçların Değerlendirilmesi:

En iyi çözüm bulunup optimizasyonu sağlanıncaya kadar iterasyon sürer. Problem için tanımlanan durdurma kriteri sağlana kadar bu şekilde çalışmaya devam eder.

GWO iç implementasyonunda algoritmanın performansını belirleyen 5 hiperparametre bulunmaktadır. Nüfus Büyüklüğü(population\_size), iterasyon sayısı(max\_iterations), alfa parametresi aralığı (a\_max, a\_min), Levy uçuş parametresi (beta). Alfa parametresi başta geniş arama sonraki iterasyonlarda dar yakınsamayı sağlamaktadır. Beta lokal minimumdan kaçmayı kolaylaştırıp algoritmanın kısır döngüye girmesini engeller. Mirjalili vd. (2014) çalışmalarında hiperparametre değerleri için aşağıdaki listeyi önermiştir: (Mirjalili vd. 2014)

- Nüfus Büyüklüğü: 5-30 (Pratikte 20-50)
- İterasyon Sayısı: 50-500

- A\_max: 2.0
- A\_min: 0.0 (lineer azalan)
- Beta: 1.5-2.0

Literatür referans alınarak Grid Search yaklaşımı uygulanarak GWO hiperparametre değerleri hesaplanmıştır. Uygulama sonucu aşağıdaki değerler elde edilmiştir:

- Nüfus Büyüklüğü: 30
- İterasyon Sayısı: 100
- A\_max: 2.0
- A\_min: 0.0 (lineer azalan)
- Beta: 1.5

DeneySEL sonuçların literatür ile örtüştüğü görülmüştür. Nihai olarak nüfus büyüklüğünün diğer optimizasyonlar ile uyumlu olması için 100 olmasına karar verilmiştir. GWO'ya ait sonuçları Test Sonuçları bölümünde verilecektir.

### 3.4.7 Gri kurt optimizasyonu (GWO) — MEC aktarım probleminde uygulama adımları

GWO adımları, Bölüm 3.4'teki ortak F(x) çerçevesi ve değerlendirme protokolü üzerinde yürütülür; karar değişkeni x (otuz iki boyutlu karar değişkeni vektörü), uygunluk simülasyon çıktısıdır (Bölüm 3.4.4).

GWO'da  $\alpha$ ,  $\beta$  ve  $\delta$  kurtları en iyi üç çözümü;  $\omega$  kurtları arama uzayında güncellenmektedir. t iterasyonunda sönümlenme parametresi:

$$a(t) = a_{\max} - (a_{\max} - a_{\min}) \cdot \frac{t}{T} \quad (3.11)$$

d boyutu için av konumu bileşenleri (Mirjalili vd. 2014; tez uygulaması):

$$A_k = a(t) \cdot (2r_k - 1), C_k = 2r'_k, r_k, r'_k \sim U(0,1), k \in \{\alpha, \beta, \delta\} \quad (3.12)$$

$$X_{k,d} = X_{k,d} - A_k \cdot |C_k \cdot X_{k,d} - X_{\omega,d}^t| \quad (3.13)$$

Ağırlıklı birleşim ve Levy adımı:

$$\eta \sim U(0,1); w_1 = 0,4\eta, w_2 = 0,35\eta, w_3 = 0,25\eta \quad (3.14)$$

$$L(\beta) = 0,01 \cdot \frac{u}{|v|^{1/\beta}}, u, v \sim N(0,1), \beta = 1,5 \quad (3.15)$$

$$X_{\omega,d}^{t+1} = w_1 X_{\alpha,d} + w_2 X_{\beta,d} + w_3 X_{\delta,d} + L(\beta) \quad (3.16)$$

Sunucu kaynak öz niteliklerinde (CPU/bellek kullanılabilirliği) ağırlıklı ortalama  $(X_{\alpha,d} + X_{\beta,d} + X_{\delta,d})/3$  kullanılmaktadır. Güncelleme sonrası  $x$ , simülatör sınır aralıklarına kırılmaktadır.

Denklem (3.14) – (3.16) içindeki  $w_1$ ,  $w_2$  ve  $w_3$ , GWO konum birleştirmesinde  $\alpha$ ,  $\beta$  ve  $\delta$  kurtlarının ağırlık katsayılarını ifade eder; bu  $w$  sembolleri amaç fonksiyonundaki  $w_1=0,7$  ve  $w_2=0,3$  ağırlıklarından (Denklem 3.23) ayrıdır.

- 1) Alfa-beta-delta kurtlar en iyi üç görevi temsil etmektedir (Mirjalili vd. 2014).
- 2)  $\omega$  kurtları  $\alpha$ – $\beta$ – $\delta$  konumlarına göre güncellenmekte;  $a(t)$  parametresi 2,0'dan 0,2'ye lineer azalmaktadır ( $a_{\min}=0,2$ ).
- 3) Yüksek boyutlu profillerde agresif arama.
- 4) Düşük bağlantı hızlı profillerde popülasyon dağılımı bozulabilir.

Literatür uyumu: Mirjalili vd. (2014).

### 3.4.8 Karınca kolonisi optimizasyonu (Ant Colony Optimization (ACO))

Karınca Kolonisi Optimizasyonu (Ant Colony Optimization- ACO), grafikler (graphs) üzerinde iyi yollar bulma problemine indirgenebilen hesaplama problemlerini çözmek için kullanılan olasılıksal bir tekniktir. Bu yöntem, Marco Dorigo tarafından 1990'ların başında doktora tezi kapsamında geliştirilmiştir. (Dorigo 1992) ACO, gerçek karınca kolonilerinin yiyecek arama davranışından esinlenmiştir. Gerçek karıncalar, yiyecek kaynaklarına giden en kısa yollar hakkında bilgi iletmek için feromon izleri bırakırlar.

Yapay karıncalar bu davranışı taklit ederek çözümler oluşturmaktadır ve sonraki arama süreçlerine rehberlik ederler.

ACO algoritması bünyesinde 6 adet hiperparametre bulundurmaktadır. Karınca sayısı (population\_size), iterasyon sayısı (max\_iterations), feromon önemi(alpha), heuristik önemi(beta), feromon buharlaşma oranı(rho), feromon miktarı (Q). Feromon matrisi çözümlerin kalitesini tutar, iyi yolları güçlendirir. Heuristik iyi çözümlere yönelimi cesaretlendirir. Buharlaşma eski yolların unutulmasını sağlayarak çeşitliliği korur. Literatürde parametre değerleri aşağıdaki gibi önerilmektedir (Dorigo vd. 1997):

- Karınca Sayısı: 10-50
- Alpha: 1.0
- Beta: 2.0-5.0
- Rho: 0.1-0.5
- Q: 10-100

Tez kapsamındaki problem uzayına göre Random Search kullanılarak parametre değerleri hesaplanmış ve literatür ile karşılaştırılmıştır. Buna göre aşağıdaki değerler elde edilmiştir:

- Karınca Sayısı: 50
- Alpha: 1.0
- Beta: 3.0
- Rho: 0.3
- Q: 50

DeneySEL sonuçlar ile literatürün örtüştüğü görülmüştür. Nüfus büyüklüğü genel normlara uyumlu olması açısından 100 olarak seçilmiştir. Algoritmanın sonuç performansına etkisi doğrudan olmayıp sadece hesaplama süresini artırmıştır. ACO'ya ait sonuçları Test Sonuçları bölümünde verilecektir.

### 3.4.9 Karınca kolonisi optimizasyonu (ACO) — MEC aktarım probleminde uygulama adımları

ACO adımları aynı değerlendirme protokolünü kullanır: her aday görev için  $F(x)$  hesaplanmaktadır, feromon güncellemesi düşük maliyetli yürütme yoluna yönlendirilir (Bölüm 3.4; Dorigo vd. 2006).

ACO uygulamasında  $N=100$  karınca ve  $N \times N$  feromon matrisi  $\tau$  kullanılmaktadır. Başlangıçta  $\tau_{ij}=1$  ( $i \neq j$ ).  $i$  indeksli başlangıç çözümünden  $j$  popülasyon üyesine geçişte öznitelik grubu  $g$  için seçim olasılığı:

$$P_{ij} = (\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta \quad (3.17)$$

$\eta_{ij} = 1 / (1 + \text{sim}(x_i, x_j))$ ;  $\text{sim}(\cdot)$  dört temel öznitelik (mesafe, arazi tipi, CPU ve bellek kullanılabilirliği) üzerinden normalize farkla hesaplanmaktadır.  $r \sim U(0,1)$  için  $r < P_{ij}$  olduğunda ilgili öznitelik grubu  $j$  çözümünden kopyalanmaktadır.

Her iterasyon sonunda buharlaşma ve depozit:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}, \rho = 0,3 \quad (3.18)$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_k \frac{Q}{F(x_k)} \cdot \text{sim}(x_k, x_j), Q = 50 \quad (3.19)$$

$F(x_k)=\infty$  olan çözümler depozite katkı vermemektedir. Oluşturulan aday görev,  $F(x)$  ile değerlendirilip sınır aralıklarına kırılmaktadır.

1)  $N=100$  karınca; feromon matrisi  $\tau$  öznitelik grubu tercihlerini kodlamaktadır (Dorigo vd. 2006).

2)  $\text{Alpha}=1$ ,  $\text{beta}=3$ ,  $\text{rho}=0,3$ ,  $Q=50$ .

3) Orta kaynak senaryolarında yakınsama gecikmesi gözlemlenebilir.

4) En iyi karınca yolu görev olarak simülatöre gönderilir.

Literatür uyumu: Dorigo vd. (1997); Dorigo vd. (2006).

### 3.4.10 Yapay arı kolonisi (Artificial Bee Colony)

Yapay Arı Kolonisi (Artificial Bee Colony- ABC) algoritması, bal arısı sürülerinin yiyecek arama davranışından esinlenen, popülasyon tabanlı bir optimizasyon algoritmasıdır. Bu algoritma, 2005 yılında Derviş Karaboğa tarafından geliştirilmiştir. Arıların yiyecek ararken izledikleri yolları ve yiyecek kaynakları hakkında bilgi paylaşımını temel almaktadır. (Karaboğa 2005) Genellikle sayısal optimizasyon problemlerini çözmek için önerilmektedir. ABC iç implementasyonunda 3 adet hiperparametre kullanılmaktadır: Arı sayısı (population\_size), iterasyon sayısı (max\_iterations) ve limit. Limit parametresi çeşitliliği korur ve lokal minimumda çakılı kalmayı engeller. Bir çözümün iyileştirilememesi durumunda yeni bir çözümle değiştirilmesini sağlamaktadır. Arı nüfusu işçi, gözcü ve izci arılardan oluşur. İşçi arılar mevcut çözümleri geliştirir, gözcüler olasılığa göre yeni çözümler dener, izci arılar ise başarısız çözümleri rastgele yeniler. Karaboğa hiperparametreler için aşağıdaki önerileri yapmıştır (Karaboğa 2005):

- Arı Sayısı: 20-100
- Limit: 40-200

Tez kapsamındaki problem uzayına göre Grid Search kullanılarak parametre değerleri hesaplanmıştır ve aşağıdaki değerler elde edilmiştir:

- Arı Sayısı: 100
- Limit: 50

Deneysel sonuçlar ve literatür birbiriyle örtüşmektedir. Tez kapsamında ABC için ulaşılan sonuçlar test sonuçları bölümünde verilecektir.

### 3.4.11 Yapay arı kolonisi (ABC) — MEC aktarım probleminde uygulama adımları

ABC adımlarında da karar uzayı  $x$  ve skor  $F(x)$  ile tanımlıdır; koloni araması yürütme yolu kıyasını değiştirmez, yalnızca aday üretim stratejisini belirlemektedir (Karaboga vd.

2007). ABC uygulamasında koloni boyutu  $N=100$  ve scout limiti  $L=50$  kullanılmaktadır. İşçi arı fazında  $i$  çözümü için rastgele komşu  $j$  ( $j \neq i$ ) seçilmekte; yeni aday  $x'_i$ ,  $x_i$  ve  $x_j$ 'nin öznitelik gruplarından (coğrafi, görev, yerel, sunucu, ağ, dış ortam) bağımsız olarak %50 olasılıkla kopyalanmasıyla üretilmektedir.

Açgözlü kabul kuralı:

$$x_i^{t+1} = \begin{cases} x'_i & F(x'_i) < F(x_i^t) \\ x_i^t & \text{aksi halde} \end{cases} \quad (3.20)$$

iyileşme olmazsa deneme sayacı  $trial\_i$  bir artırılmaktadır. Gözcü arı fazında seçim olasılığı:

$$p_i = \frac{f_i}{\sum_k f_k}, f_i = \frac{1}{1+F(x_i)} \quad (3.21)$$

Scout fazında  $trial\_i \geq L$  olduğunda  $x_i$  yeni rastgele görev ile yenilenmekte ve  $trial\_i$  sıfırlanmaktadır.

- 1) İşçi arılar mevcut komşuluk araması ile iyileştirir (Karaboga vd. 2007).
- 2) Gözcü arılar olasılıklı seçim yapar.
- 3) İzci arılar limit=50 sonrası başarısız çözümleri yeniler.
- 4) Düşük kaynak senaryolarında kaynak verimliliği üstün gelebilir.

Literatür uyumu: Karaboga vd. (2007); Karaboğa (2005).

Sezgisel algoritmaların performansı için hiperparametreler büyük önem taşımaktadır. Algoritmalar bu parametreler aracılığıyla farklı problem uzaylarına uygulanabilir olmaktadır. Optimum değerlere ulaşabilmek için hem deneysel hem de akademik çalışmalar birlikte kullanılmıştır. Tüm algoritmalar için hiperparametre değerleri ve akademik öneriler Çizelge 3.8'te özetlenmiştir:

### 3.4.12 Meta-sezgisel algoritmaların MEC aktarım problemine uygulanması

Bu alt bölümde, Bölüm 3.4'te tanıtılan GA, PSO, GWO, ACO ve ABC'nin karar değişkeni vektörü  $x$  üzerindeki arama uzayı, kodlama ve  $F(x)$  bağlantısı özetlenmektedir. Ortak çerçeve: popülasyon boyutu 100, iterasyon 100, batch 500 (Poisson senaryo deneyleri); gerçek dünya veri setleri (Bölüm 4.1.3).

Tüm algoritmalarda Poisson senaryo deneylerinde popülasyon 100 ve iterasyon 100; gerçek dünya deneylerinde task\_factory değerlendirme döngüsü (Bölüm 3.5.3) ve hiperparametreleri ortak protokol olarak uygulanmıştır (Bölüm 4.1.5).

Genetik Algoritma (GA): Kromozom, otuz iki sürekli ve ayrık boyuttur. Uygunluk  $F(x)$ 'in simülatörden dönen değeridir. Düşük değerler tercih edilir. Çaprazlama iki ebeveyn özniteliklerini karıştırır. Mutasyon tek özniteliği Poisson merkezli aralıkta pertürb eder. Elitizm (elite\_size=10) en iyi çözümlerin kaybını önler. Düşük kaynak senaryolarında dengeli kaynak tüketimi sağlamaktadır (Goldberg 1989). GA, dinamik senaryolarda popülasyon çeşitliliği sayesinde daha hızlı adaptasyon gösterebilir (Bölüm 4.1.16).

Parçacık Sürüsü Optimizasyonu (PSO): Her parçacık bir görev konumudur. Hız vektörü öznitelik güncellemelerini belirlemektedir. Kişisel en iyi ve küresel en iyi  $F(x)$  kayıtları sürü davranışını yönlendirir (Kennedy vd. 1995). Yüksek yoğunluklu senaryolarda agresif keşif, kısa gecikme ile yüksek kaynak tüketimi değiş-tokuş'u (trade-off) üretmektedir. Bu sonuç Bölüm 4.1'deki GWO ile benzer örüntüdür.

Gri Kurt Optimizasyonu (GWO): Alfa, beta, delta ve omega kurtları en iyi üç çözüm etrafında avlanma güncellemesi yapar (Mirjalili vd. 2014). Levy flight benzeri geniş adımlar, yüksek görev hacmi ve kullanıcı yoğunluğu yüksek profillerinde hızlı iyileşme sağlayabilir; düşük kaynak profillerinde popülasyon dağılımı kötüleşebilir.

Karınca Kolonisi Optimizasyonu (ACO): Feromon matrisi, öznitelik alt uzaylarında

tercih edilen yönleri biriktirir (Dorigo vd. 2006). Orta kaynak senaryolarında feromon güncelleme maliyeti nedeniyle yakınsama gecikebilir; bu tezdeki ACO sonuçları bu gözlemi desteklemektedir (Bölüm 4.1.16). Alpha, beta, rho ve Q değerleri Çizelge 3.8 ile literatür aralığında hizalanmıştır.

Yapay Arı Kolonisi (ABC): İşçi, gözcü ve izci arılar limit parametresi ile sıkışmayı önler (Karaboga vd. 2007). Düşük connection\_speed ve düşük user\_density senaryolarında ABC'nin kaynak verimliliği, GA ile en güçlü grupta yer almaktadır.

Çizelge 3.8 Meta-sezgisel algoritma hiperparametreleri

| Algoritma | Deneyisel Değerler                         | Akademik Öneri                                        | Uyum   |
|-----------|--------------------------------------------|-------------------------------------------------------|--------|
| GA        | Pop:100, cross:0.8, mut:0.05, elite:10     | Pop:50-200, cross:0.6-0.9, mut:0.01-0.1, elite: %5-20 | Uyumlu |
| PSO       | Pop:50, w:0.7, c1:2.0, c2:2.0              | Pop:10-50, w:0.7-0.9, c1/c2:1.5-2.5                   | Uyumlu |
| GWO       | Pop:30, a_max:2.0, a_min:0.0               | Pop:5-30, a_max:2.0, a_min:0.0                        | Uyumlu |
| ACO       | Pop:50, alpha:1.0, beta:3.0, rho:0.3, Q:50 | Pop:10-50, alpha:1.0, beta:2-5, rho:0.1-0.5, Q:10-100 | Uyumlu |
| ABC       | Pop:100, limit:50                          | Pop:20-100, limit:100-200                             | Uyumlu |

### 3.4.13 MEC operatörü için meta-sezgisel karar döngüsü

Meta-sezgisel algoritmanın MEC'de kullanımı şu döngüyle özetlenebilir (Mach vd. 2017) (Mao vd. 2017):

- 1) Bağlam örneği: Senaryo üretici bir örnek vektörü  $x$  üretmektedir (mesafe, task\_size, connection\_speed, user\_density vb.).
- 2) Aday üretimi: Seçilen algoritma (GA/PSO/GWO/ACO/ABC) popülasyonunda  $x$  varyantları oluşturmaktadır.

- 3) Değerlendirme: Simülatör her  $x$  için yürütme yollarını hesaplar;  $F(x)$  döner (Bölüm 3.4).
- 4) Seçim: Düşük  $F(x)$  üstün kabul edilmektedir; elitizm veya gbest/alfa güncellemesi bir sonraki nesle taşınır (Goldberg 1989) (Mirjalili vd. 2014).
- 5) Karar çıktısı: En iyi  $x$ 'e karşılık gelen yürütme yolu operatör politikasını temsil etmektedir.

Fayda boyutları: (a) Tek görev düzeyinde karar — gerçek zamanlı uygulamalara yakın prototip (Taleb vd. 2017). (b) Gecikme–kaynak trade-off'un tek skorda birleştirilmesi (Rao 2009). (c) Tümü yerel / tümü uç referans çizgilerine göre iyileşmenin ölçülebilir olması (Bölüm 3.5.2). (d) Algoritma değiştirildiğinde aynı  $F(x)$  ile adil kıyas (Kennedy vd. 1995) (Karaboga vd. 2007).

Bu döngü, meta-sezgisellerin simülatör ile bağlantılı, yorumlanabilir bir MEC karar mekanizmasına dönüşmesini sağlamaktadır (You vd. 2017).

#### **3.4.14 Adil kıyas: arama limiti ve fitness değerlendirme maliyeti**

Optimizasyon makalelerinde adil kıyas için üç kural bu tezde protokolde belirlenmiştir (Deb 2002) (Eiben vd. 2015): (B1) aynı amaç fonksiyonu  $F(x)$ ; (B2) aynı popülasyon (100) ve iterasyon (100) limiti; (B3) aynı simülatör ve ikili yol değerlendirme politikası (Bölüm 3.4.4).

Her fitness değerlendirmesi, bir aday görev için yürütme yollarının simülatörde hesaplanması ve düşük  $F(x)$  yolunun seçilmesidir; dolayısıyla tek aday başına en az iki simülatör çağrısı yapılmaktadır. Üst sınır Denklem (3.12) ile verilmiştir;  $N$  popülasyon boyutu,  $T$  iterasyon sayısıdır. Pratikte değerlendirme sayısı popülasyon $\times$ iterasyon mertebesinde kalmaktadır. Beş algoritma aynı  $N\_eval$  limitine sahip olduğundan, sıralama farkları limit adaletsizliğinden değil operatör farkından kaynaklanır (Goldberg 1989) (Kennedy vd. 1995).

$$N_{eval} \leq 2 \cdot N \cdot T \quad (3.22)$$

Gerçek veri seti deneylerinde havuz boyutu değişse de iyileştirici (optimizer) başına popülasyon/iterasyon sabittir; tekrarlı koşullarda (Bölüm 4.1.9) yalnızca seed değişmektedir, limit değişmez (Rao 2009).

### 3.4.15 Yakınsama, global optimum ve stokastik sınırlar

Meta-sezgisel yöntemler polinom zamanlı iterasyon yapar. Ancak MEC aktarım problemi çok modlu ve stokastik başlangıç popülasyonu nedeniyle global optimum garantisi vermemektedir (Wolpert vd. 1997). No Free Lunch teoremi, tüm olası problemler üzerinde tek bir algoritmanın sürekli üstün olamayacağını ifade etmektedir. Bu tezdeki AS1 bulgusu (veri seti/senaryo başına farklı lider) bu çerçeveye uyumludur.

Yakınsama yorumu: düşük  $F(x)$  veya tr bandına erken oturma, global optimum kanıtı değildir. seed değiştikçe en iyi çözüm değişebilir (Bölüm 4.1.9). Bu nedenle tez, “en iyi bulundu” iddiasını tekrarlı ortalama $\pm$ std ve isteğe bağlı Wilcoxon ile sınırlar (Wilcoxon 1945) (Deb 2002). Pareto ön yüzü raporlanmamıştır. Ağırlıklı toplam tercih edilmiştir (Bölüm 3.5.2) (Mao vd. 2017).

### 3.5 Amaç (Objective) Fonksiyonu

Karar değişkeni vektörü  $x$ , bir MEC görev örneğinin görev yükü, yerel cihaz, ağ, sunucu ve çevre özniteliklerini içeren otuz iki değişkenli sayısal kodlamasıdır. Uygulamada  $x$ , Task nesnesi biçiminde temsil edilmektedir. Amaç fonksiyonu  $F(x)$  Denklem (3.1) ile tanımlanmış olup  $f_1$  gecikme,  $f_2$  kaynak tüketimi bileşenlerini göstermektedir. Her  $x$  için simülatör yerel ve uç yürütme yollarını ayrı değerlendirmekte; daha düşük  $F(x)$  değerini veren yol seçilmektedir.

Meta-sezgisel karşılaştırmada amaç fonksiyonunun açık ve tutarlı tanımlanması gerekir. Rao (2009), ölçek uyumu, çok kriterli birleştirme ve minimizasyon yönünün açıkça

belirtilmesini önerir. Bu tezde hesaplama aktarımı minimize  $F(x)$  problemi olarak formüle edilmiştir.

Optimizasyon problemi  $\min_{\{x \in X\}} F(x)$  biçiminde tanımlanmıştır. Burada  $X$ , TARGET\_BOUNDS ile sınırlandırılmış otuz iki boyutlu karar uzayıdır (Bölüm 3.5.4). Her aday  $x$  için simülatör yerel (LOCAL) ve uç (EDGE) yürütme yollarını ayrı değerlendirmekte; daha düşük  $F(x)$  değerini veren yol aktarım kararı olarak seçilmektedir (Bölüm 3.4.4).

Çok kriterli ağırlıklı toplam yaklaşımında Denklem (3.23) kullanılmıştır.  $f_1(x)$  gecikme,  $f_2(x)$  kaynak bileşenini temsil etmektedir; total\_delay(x) Denklem (3.26) ile delay(x) ve execution(x) olarak ayrıştırılmıştır. resource(x) bileşen ağırlıkları Denklem (3.6) ile verilmiştir. Ağırlıklar Denklem (3.27) ile sabittir; gecikme birincil, kaynak ikincil kriterdir. Sembol tanımları Çizelge 3.9'de özetlenmiştir.

Not: Denklem (3.24) ve Denklem (3.26) içindeki total\_delay, delay ve execution terimleri milisaniye (ms) cinsindendir;  $f_1$  ve  $f_2$  boyutsuz ölçeklenmiş bileşenlerdir.

$$F(x) = w_1 f_1(x) + w_2 f_2(x) \quad (3.23)$$

$$f_1(x) = \max\left(\frac{\text{total\_delay}(x)}{100}, 1\right) \quad (3.24)$$

$$f_2(x) = \max\left(\frac{\text{resource}(x)}{100}, 1\right) \quad (3.25)$$

$$\text{total\_delay}(x) = \text{delay}(x) + \text{execution}(x) \quad (3.26)$$

$$w_1 = 0,7, w_2 = 0,3 \quad (3.27)$$

Rao (2009) ilkeleri ile tezdeki  $F(x)$  tanımının uyumu Çizelge 3.10'de özetlenmiştir.

Denklem (3.23) – (3.32) ve Bölüm 4 raporlama kısaltmalarında kullanılan temel

semboller Çizelge 3.9'de özetlenmiştir.

Çizelge 3.9 Amaç fonksiyonu semboller ve tanımları

| Sembol                   | Tanım                                                                       | Birim  |
|--------------------------|-----------------------------------------------------------------------------|--------|
| $x$                      | Otuz iki boyutlu karar vektörü<br>(Task kodlaması)                          | —      |
| $X$                      | TARGET_BOUNDS ile<br>sınırlandırılmış karar uzayı                           | —      |
| $F(x)$                   | Ağırlıklı toplam amaç<br>fonksiyonu (Denklem 3.1)                           | —      |
| $f_1(x)$                 | Gecikme bileşeni:<br>$\max(\text{total\_delay}(x)/100, 1)$<br>(Denklem 3.2) | —      |
| $f_2(x)$                 | Kaynak bileşeni:<br>$\max(\text{resource}(x)/100, 1)$<br>(Denklem 3.3)      | —      |
| $w_1, w_2$               | Amaç ağırlıkları (0,7; 0,3)<br>(Denklem 3.5)                                | —      |
| $\text{total\_delay}(x)$ | $\text{delay}(x) + \text{execution}(x)$<br>(Denklem 3.4)                    | ms     |
| $\text{delay}(x)$        | İletim ve kuyruk gecikmesi<br>(Denklem 3.7–3.8; yol<br>seçimine bağlı)      | ms     |
| $\text{execution}(x)$    | İşlem süresi (Denklem 3.9)                                                  | ms     |
| $\text{resource}(x)$     | Çoklu kaynak kullanım skoru<br>(Denklem 3.6)                                | %      |
| $N, T$                   | Popülasyon boyutu ve<br>iterasyon sayısı                                    | —      |
| $N_{\text{eval}}$        | Üst sınır fitness<br>değerlendirme sayısı<br>(Denklem 3.12)                 | —      |
| $tr, td, te, cr$         | Raporlanan toplam yanıt,<br>iletim, yürütme süreleri ve<br>kaynak (Bölüm 4) | ms / % |

Çizelge 3.10 Rao (2009) ilkeleri ile  $F(x)$  tanımının uyumu

| Rao (2009) ilkesi            | Tezdeki karşılık                                                                                                      |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Performans ölçütü            | $\text{total\_delay}(x)=\text{delay}(x)+\text{execution}(x)$ ve $\text{resource}(x)$ bileşenlerinin minimize edilmesi |
| Normalizasyon ve birim uyumu | Her kriter 100'e bölünür; sonuç 1'in altına düşerse 1 alınır (Denklem 3.2 ve 3.3)                                     |
| Çok kriterli birleştirme     | Ağırlıklı toplam; $w_1=0,7$ , $w_2=0,3$                                                                               |
| Minimizasyon yönü            | Düşük $F(x)$ değeri daha iyi çözümü gösterir                                                                          |
| Süreklilik                   | Task vektörü üzerinde sürekli; türevlenebilirlik sezgisel arama için şart değildir                                    |
| Kısıtlar                     | TARGET_BOUNDS kırpm ve $\max(\cdot, 1)$ alt sınırı (Bölüm 3.5.4)                                                      |

$F(x)$  tanımı Rao (2009) ilkeleriyle uyumlu olacak biçimde kurgulanmış ve tüm algoritma, senaryo ile veri seti deneylerinde değiştirilmeden kullanılmıştır.

### 3.5.1 Çok kriterli amaç fonksiyonunun formülasyonu ve ağırlıklandırma

Rao (2009), amaç fonksiyonunun ölçek uyumu, ağırlıklı toplam ve minimizasyon yönü açısından açık tanımlanmasını önerir. Bu tezde  $F(x)$  seçimi gecikmeyi birincil ( $f_1$ ), kaynak kullanımını ikincil ( $f_2$ ) kriter olarak kodlamaktadır. Ağırlıklar tüm algoritma ve senaryolarda sabittir. Böylece Bölüm 4.1.16'teki AS2 kapsamında gözlemlenen farklar esasen operatör kaynaklı yorumlanır. Ağırlık değişikliği meta-sezgisel değil deney protokolü değişikliği olurdu.  $f_1$  ve  $f_2$  simülatör formüllerinden türetilmektedir; saha ölçümü değildir (Bölüm 3.4.2) (Mao vd. 2017). Gerçek dünya veri seti deneylerinde ham sonuç sütunları önce standart öznitelik şemasına yansıtılır, ardından aynı  $F(x)$  uygulanır (Guo vd. 2019) (Hadary vd. 2020). Bu birleşik skor, Poisson senaryoları ile gerçek dünya veri setleri arasında AS3'ün (Bölüm 4.1.16) test zemini olarak işlev görür.

### 3.5.2 Gecikme ve kaynak metriklerinin simülatör çıktısı olarak türetilmesi

Bu çalışmada gecikme ve kaynak kullanımı saha ölçümüyle toplanmamıştır.  $\text{delay}(x)$ ,  $\text{execution}(x)$  ve  $\text{resource}(x)$  karar değişkeni vektörü  $x$  üzerinde simülatör formülleriyle

hesaplanmıştır (Denklem 3.28, 3.29–3.33). Bu tercih, tüm algoritmalar ve referans çizgisi için aynı fiziksel modeli zorunlu kılar (Mao vd. 2017) (Mach vd. 2017). Simülasyon gerçek dünyanın tüm belirsizliğini taşıyamaz; ancak yöntemler arası görelî sıralama ve senaryo bazlı eğilimler için ortak zemin sağlar (Rao 2009).

Toplam gecikme ayrıştırması Denklem (3.28)'te verilmiştir.  $\text{delay}(x)$  ve  $\text{execution}(x)$  alt bileşenleri, UÇ ve YEREL yürütme yollarına göre Denklem (3.28) – (3.32) ile hesaplanmaktadır.

UÇ yolunda iletişim gecikmesi aşağıdaki gibidir.

$$\text{delay}(x) = \text{effectivedistance}(x) / \text{commspeed}(x) \cdot 10^3 \cdot \text{queueeffect}(x) \quad (3.28)$$

YEREL yolunda iletişim gecikmesi aşağıdaki gibidir.

$$\text{delay}(x) = 10^3 \cdot \text{queueeffect}(x) \cdot \text{resourceeffect}(x) \quad (3.29)$$

$$\text{execution}(x) = \text{bulktasksize}(x) / \text{processingcapacity}(x) \cdot \text{queueeffect}(x) \cdot 10^3 \quad (3.30)$$

$$\text{queueeffect}(x) = 1 + \text{networkqueue}(x) / 100 \quad (3.31)$$

$$\text{resourceeffect}(x) = 1 + (100 - \text{cpuavailability}(x)) / 200 \quad (3.32)$$

$\text{queueeffect}$  ve  $\text{resourceeffect}$  çarpanları sırasıyla Denklem (3.31) ve (3.32) ile tanımlanmıştır.  $\text{effectivedistance}$ ,  $\text{bulktasksize}$  ve  $\text{processingcapacity}$  terimleri ile  $\text{recurrenceeffect}$  çarpanı Task vektörü özniteliklerinden türetilmektedir (Çizelge 3.2–3.6).

$\text{resource}(x)$  Denklem (3.33) ile tanımlanmış olup  $\text{CPU\_usage}(x)$ ,  $\text{MEM\_usage}(x)$ ,  $\text{storage\_usage}(x)$ ,  $\text{network\_usage}(x)$  ve  $\text{bus\_usage}(x)$  yüzdelerinin ağırlıklı toplamına karşılık gelmektedir. Bu alanlar Task şemasındaki yerel ve uç sunucu öznitelikleriyle eşleştirilmiştir (Çizelge 3.2–3.6). Yanıt süresi (response time) raporları milisaniye (ms) cinsindendir; mutlak değerlerden çok algoritma ve senaryo görelî farkları yorumlamak

gerekir. Kaynak kullanımı yüzdesi normalize alt bileşenlerin ağırlıklı toplamından türetilir; üst sınır 100 olarak Bölüm 3.4'te belirtilmiştir. Gerçek dünya veri setlerinde öznitelik projeksiyonu sonrası aynı  $F(x)$  uygulanır (Bölüm 3.5.3) (Guo vd. 2019) (Hadary vd. 2020).

### 3.5.3 Amaç fonksiyonlarının analitik değerlendirilmesi

$f_1(x)$  ve  $f_2(x)$  sırasıyla Denklem (3.24) ve Denklem (3.25) ile tanımlanır.  $\text{delay}(x)$  ve  $\text{execution}(x)$  alt bileşenleri Denklem (3.28), (3.30) ile;  $\text{resource}(x)$  Denklem (3.33) ile hesaplanır.

Amaç fonksiyonu gecikmeye öncelik verir: aynı kaynak tasarrufunda bile gecikme düşüşü skoru daha çok düşürür (Rao 2009). 100 paydası mutlak milisaniye değerlerini ölçekler; bu nedenle Bölüm 4'teki ms değerleri görelî kıyas içindir, saha ölçümü değildir.

Örnek yorum: bir senaryoda PSO/GWO eğrisi  $f_1$  açısından aşağıda, GA/ABC  $f_2$  açısından daha dengeli ise, Çizelge 4.4'teki önerilerle uyumlu bir profil okuması yapılmış olur (Kennedy vd. 1995) (Goldberg 1989).

Aktarım kararında aynı  $x$  için UÇ ve YEREL yürütme yolları ayrı değerlendirilir; daha düşük  $F(x)$  veren mod seçilir (Bölüm 3.4.4). Denklem (3.6) içindeki  $\text{CPU\_usage}(x)$ ,  $\text{MEM\_usage}(x)$ ,  $\text{storage\_usage}(x)$ ,  $\text{network\_usage}(x)$  ve  $\text{bus\_usage}(x)$  terimleri Task vektöründeki ilgili kullanım yüzdelerine karşılık gelmektedir; ağırlıklar (0,3; 0,3; 0,15; 0,15; 0,1) çoklu tüketimi tek skalerde birleştirir (Mao vd. 2017).

$$\text{resource}(x) = 0,3 \cdot \text{CPU} + 0,3 \cdot \text{MEM} + 0,15 \cdot \text{STO} + 0,15 \cdot \text{NET} + 0,1 \cdot \text{BUS} \quad (3.33)$$

### 3.5.4 Karar uzayı, kısıtlar ve problem karmaşıklığı

Karar değişkeni  $x$ ; sürekli ve ayrık karışık bir uzaydır (görev hacmi, bağlantı hızı, sunucu tipi vb.). Her aday için yerleşim YEREL veya UÇ olarak simülatörde değerlendirilir; bu

ikili seçim arama uzayına kombinatoriyal bir boyut ekler (Cao vd. 2019) (Mach vd. 2017). Hesaplama aktarımı literatüründe NP-zor alt problemler yaygındır; bu tez tam topoloji optimizasyonu yapmaz, sabit  $F(x)$  altında  $x$  araması yapar (Taleb vd. 2017).

Kısıtlar iki katmandadır: (Kısıt-1) adaptör TARGET\_BOUNDS ile sayısal kırpma (clamp\_to\_simulator\_bounds); (Kısıt-2)  $F(x)$  içindeki  $\max(\cdot, 1)$  normalizasyon alt sınırları (Bölüm 3.4). Ceza fonksiyonu yerine kırpma tercih edilmiştir; sınır dışı adaylar simülatör istikrarı için elenmez, sınırdaki tutulur (Hadary vd. 2020) (Ben-David vd. 2010).

Optimizasyon çıktısı seçilen  $x^*$  ve buna bağlı yerleştirme ile gecikme/yürütme bileşenleridir (Bölüm 1.2; Bölüm 1.1.1).

### 3.6 Test Senaryoları ve Başarı Metrikleri

Testlerin daha gerçekçi ve kapsayıcı olması amacıyla yalnızca tek bir temel senaryoyla yetinilmemiş; farklı koşulları temsil edebilecek bir dizi test senaryosu da titizlikle tasarlanarak simülasyon ortamına entegre edilmiştir. Bu kapsamda, gerçek dünya uygulamalarında karşılaşılabilecek çeşitli durumları yansıtabilmek adına, Poisson dağılımı tabanlı modelleme tercih edilmiş ve dağılımın parametreleri üzerinde sistematik varyasyonlara gidilmiştir. Böylece, olayların gerçekleşme sıklığını etkileyen faktörler kontrollü biçimde değiştirilerek sistemin farklı yük ve koşullar altındaki performansı gözlemlenebilmiştir. Söz konusu çeşitlendirme sonucunda, temel (baz) senaryo dahil olmak üzere toplamda 15 farklı test senaryosu oluşturulmuş ve bu senaryolar aracılığıyla yöntemlerin yalnızca ideal şartlar altında değil, aynı zamanda değişken ve potansiyel olarak stresli koşullar altında da nasıl davrandığı incelenmiştir. Bu yaklaşım hem simülasyonun temsil gücünü artırmakta hem de geliştirilen yöntemlerin genellenebilirliğini değerlendirmek açısından önemli katkı sağlamaktadır.

#### 3.6.1 Senaryo sınıflandırması: Statik/Dinamik ve kaynak profili

On beş senaryo, iki eksenle sınıflandırılabilir. Birinci eksen — kaynak ve yoğunluk: (A)

düşük kaynak / düşük bant (Sınırlı İnternet Kırsal, Tarımsal Ağ, Geçici Ağ). (B) orta/dengeli (Baz, Banliyö, Evden Çalışma). (C) yüksek kaynak / yüksek yoğunluk (Yüksek Yoğunluklu Kentsel, Etkinlik Trafik, Büyükşehir Veri Merkezi). İkinci eksen — dinamiklik: Felaket Kurtarma ve kısmen Akıllı Şehir/Etkinlik profilleri değişken yük altında dinamik kabul edilmektedir; diğerleri statik veya yarı-statik profildir (Bölüm 4.1, Şekil 4.35–4.36). Bu taksonomi, 3.5'teki sözel senaryo tanımlarını sayısal  $\lambda$  profilleriyle (Bölüm 3.1.1) birleştirir; okuyucu açısından senaryo–sonuç eşlemesini kolaylaştırır. Detaylı tablolar Ek-B’de konumlandırılmıştır.

**Baz (Temel) Senaryo:** Bu senaryo, çalışmanın temel referans noktası olarak yapılandırılmış olup, diğer tüm senaryoların karşılaştırılmasında bir kıyas ölçütü görevi görmektedir. Gerçekçi ancak nötr bir senaryo kurgulanarak, sistemin ortalama koşullar altındaki performansını gözlemlemek amaçlanmıştır. Bu kapsamda, görev boyutları, ağ kapasitesi, kullanıcı yoğunluğu ve kaynak dağılımı gibi parametreler “orta seviye” olarak tanımlanmış ne yüksek ne de düşük yoğunluklu bir kullanım profili tercih edilmiştir. Böylece, sistemin nominal koşullarda gösterdiği davranışlar temel alınarak diğer özel durum senaryolarıyla sağlıklı bir karşılaştırma yapılabilmesi hedeflenmiştir.

**Yüksek Yoğunluklu Kentsel Alan:** (Şekil 4.3–4.4): Yüksek  $\lambda$  ve yoğun görev varışı profilinde GWO gecikme ve kaynak dengesinde en düşük değerleri üretmiş, PSO yakın performans vermiştir. GA yaklaşık %20 daha yüksek gecikme değerleri üretmiş; ABC ve ACO daha düşük sıradadır. Referans çizgilerine göre PSO/GWO bandı ölçülebilir iyileşme göstermiştir. Agresif keşif operatörleri yüksek yoğunlukta kaynak harcayarak gecikmeyi düşürmektedir; bu bulgu AS2 ile uyumludur (Kennedy vd. 1995; Mirjalili vd. 2014).

**Akıllı Şehir Ağı:** (Şekil 4.5–4.6): IoT cihazlarının heterojen talep profilinde GWO ve PSO gecikmede en düşük değerleri üretmiştir; GA, ABC ve ACO yakın ancak daha yüksek gecikme değerleri üretmiştir. Kaynak kullanımında GA ve ABC küçük bir marjla daha verimli olmuştur. Dual-path değerlendirme, yoğun edge yükünde uç seçimine izin vermiştir (Mao vd. 2017). Yüksek bağlantı çeşitliliği profillerinde sürü tabanlı keşif, feromon tabanlı aramaya göre daha hızlı yakınsama eğilimi göstermiştir (Mach vd. 2017).

**Tüketici Ağı Kullanımı Sınırlı İş Bölgesi:** (Şekil 4.7–4.8): Orta-düşük kaynak ve sınırlı tüketici bant genişliği profilinde ABC iş yükü optimizasyonunda bu koşuda en düşük gözlemlenen gecikmeyi üretmiştir. PSO, GA ve GWO yakın ancak ABC'ye göre daha yavaş yakınsamıştır; ACO orta sıradadır. Kaynak kullanımında PSO ve GWO en yüksek tüketimi üretmiştir. Limit tabanlı ABC operatörü, kısıtlı profilde elitizmden daha dengeli davranmıştır (Karaboga vd. 2007).

**Etkinlik Kaynaklı Yüksek Trafik:** (Şekil 4.9–4.10): Ani yük artışı ve dinamik trafik profilinde GA ve ABC gecikmede en düşük değerleri üretmiştir; PSO ve GWO yüksek yoğunlukta daha yüksek gecikme değerleri üretmiş, ACO orta bantta yer almıştır. Kaynak kullanımında ABC, ACO ve GA daha düşük kaynak kullanım değerleri gözlenmiştir. Popülasyon çeşitliliği, ani yük dalgalanmalarına adaptasyonu desteklemiştir (Goldberg 1989). Bu senaryo, düşük kaynak profillerinde AS2 ile uyumlu bulgular sunmaktadır.

**Sınırlı İnternet Erişimi Olan Kırsal Bölge:** (Şekil 4.11–4.12): Düşük bant genişliği ve kısıtlı edge kaynağı profilinde GWO ve PSO daha yüksek gecikme değerleri üretmiştir. ABC, ACO ve GA daha düşük gecikme değerleri gözlenmiştir. Referans çizgilerine göre agresif keşif yapan algoritmaların riskli arama uzayı genişlemesi gecikmeyi artırmıştır. Düşük kaynak profillerinde limit ve feromon tabanlı arama görece avantaj göstermiştir (Dorigo vd. 2006; Karaboga vd. 2007).

**Tarımsal Ağ İhtiyaçları:** (Şekil 4.13–4.14): Düşük kaynak ve seyrek görev varışı profilinde GA, ABC ve ACO beklendiği üzere daha düşük gecikme değerleri üretmiştir. GWO ve PSO birbirine yakın ancak daha yüksek gecikme değerleri üretmiştir. Kaynak eğrileri de aynı sıralamayı desteklemektedir. Kırsal düşük yoğunluk profillerinde elitizm ve limit tabanlı arama, agresif keşfe göre daha güvenilir sonuç üretmiştir (Goldberg 1989).

**Fiber Optik veya 5G Altyapısına Sahip Kırsal Bölge:** (Şekil 4.15–4.16): Yüksek bant genişliği profilinde GWO ve PSO hafif önde olmakla birlikte tüm iyileştiriciler yüksek performans bandına yakınsamıştır. Algoritmalar arası fark azalmıştır; referans çizgilerine göre tüm aileler iyileşme sağlamıştır. Geniş bant, edge aktarımının cezasını

düşürdüğünden keşif operatörleri görelî avantajını kısmen kaybetmektedir (Liu vd. 2023). Bu profil, gerçek dünya 5G saha verisi bulgularıyla (Bölüm 4.1.3) tutarlıdır.

**Banliyö Ofis ve Konut Ağı:** (Şekil 4.17–4.18): Orta yoğunluk ve dengeli kaynak profilinde tüm algoritmalar birbirine yakın gecikme üretmiştir; belirgin bir lider çıkmamıştır (AS1). Kaynak kullanımında GA en düşük değerleri vermiştir. Referans çizgilerine göre iyileşme sınırlı fakat tutarlıdır. Dengeli profillerde algoritma seçimi, operatör farkından çok ince ayar ve protokol parametrelerine bağılı hale gelmektedir (Deb 2002).

**Evden Çalışma ve Aile Ağı:** (Şekil 4.19–4.20): Orta kaynak ve karışık tüketim profilinde tüm iyileştiriciler yakın sonuç vermiştir. GA ve ABC küçük bir farkla öndedir; PSO ve GWO kaynak kullanımında daha yüksek tüketim göstermiştir. Referans çizgilerine göre tüm algoritmalar kabul edilebilir iyileşme sağlamıştır. Ev içi ve hibrit çalışma yüklerinde limit tabanlı arama, agresif keşfe göre daha dengeli kaynak–gecikme dengesi kurmaktadır (Karaboga vd. 2007).

**Büyükşehirde Veri Merkezi Performansı:** (Şekil 4.21–4.22): Yüksek işlem gücü ve yoğun edge kapasitesi profilinde PSO ve GWO en düşük gecikme değerini üretmiştir. GA ve ABC birbirine yakın, ACO'ya kıyasla daha düşük gecikme gözlenmiştir. Kaynak eğrileri gecikme sıralamasıyla paralellik göstermiştir. Referans çizgilerine göre PSO/GWO bandı daha düşük bir gecikme bandında konumlanmıştır. Yüksek kaynak profillerinde agresif keşif, edge aktarımının getirisini artırmaya yöneliktir (Kennedy vd. 1995; Mirjalili vd. 2014).

**Felaket Kurtarma Testi:** (Şekil 4.23–4.24): Dinamik ortam değişikliği profilinde GA en düşük gecikme uyumunu göstermiştir; ACO en yüksek gecikme değerini üretmiştir, diğer algoritmalar bu ikisi arasında sıralanmıştır. Kaynak kullanımı gecikme ile uyumludur. Referans çizgilerine göre GA'nın görece düz eğrisi, popülasyon çeşitliliğinin dinamik koşullara adaptasyonunu desteklediğini düşündürmektedir (Goldberg 1989). Bu bulgu, dinamik senaryo grubundaki (Şekil 4.33–4.34) GA eğilimiyle örtüşmektedir.

**Uzak Bölgelerde Uç Bilişim Ağı:** (Şekil 4.25–4.26): Orta-düşük kaynak ve uzak edge profilinde GA hesaplama yükünü en dengeli biçimde dağıtmış, ABC izlemektedir. PSO ve GWO yakın performans göstermiş, ACO daha yüksek gecikme değerleri üretmiştir. Kaynak kullanımında GA ve ABC daha düşük kaynak kullanım değerleri gözlenmiştir. Uzak edge senaryolarında elitizm ve limit tabanlı arama, aşırı uç aktarımını sınırlayarak dengeli yerleşim üretmiştir (Guo vd. 2019).

**Kırsal veya Uzak Alanlarda Geçici Ağ Kurulumları:** (Şekil 4.27–4.28): Orta-düşük parametre profilinde GWO ve PSO daha yüksek gecikme değerleri üretmiştir. GA, ABC ve ACO birbirine yakın ve daha düşük gecikme değerleri üretmiştir. Referans çizgilerine göre agresif keşif, geçici ve kısıtlı altyapıda riskli arama uzayı genişlemesi nedeniyle daha yüksek gecikme değerleri üretmiştir. Düşük kaynak geçici kurulumlarda GA/ABC/ACO önceliği AS2 ile uyumludur.

**Kırsal Alanlarda Teletıp ve E-Öğrenme:** (Şekil 4.29–4.30): Yüksek ağ hızı ve yeterli kaynak profilinde tüm iyileştiriciler birbirine yakın gecikme üretmiştir; belirgin lider çıkmamıştır. Kaynak eğrileri de benzer yakınsama göstermiştir. Geniş bant, edge aktarım cezasını düşürdüğünden operatör farkları azalmaktadır. Bu profil, Fiber/5G kırsal senaryosu ve 5G saha verisi bulgularıyla (Bölüm 4.1.3) tutarlı bir "yüksek bant – düzleşmiş sıralama" örüntüsü sunmaktadır (Liu vd. 2023).

Tüm bu senaryolar, sistemin farklı koşullar altındaki performansını bütüncül bir bakış açısıyla değerlendirebilmek için tasarlanmış olup, kıyaslamalı analizler açısından zengin bir test ortamı sunmaktadır. Simülasyon testleri, beş karar verici algoritma (GA, PSO, GWO, ACO, ABC) için 1000’lik kümelerden oluşan 500 grup halinde yapılmıştır. Gecikme (Delay) ve kaynak kullanımı (Resource Consumption), başarı ölçütleri olarak seçilmiştir. Gecikme bir görevin veya işlemin yapılması için geçen toplam süreyi temsil etmektedir. Kaynak kullanımı ise o görevin yapılması için geçen sürede sistem kaynaklarının yüzde kaçının kullanılması gerektiğini belirtir. Gecikme birim zaman olarak hesaplanırken, kaynak kullanımı genel sistem kullanımının yüzdesi olarak belirlenir. MEC gecikme problemine çözüm olarak geliştirildiği için gecikmenin ağırlığı kaynak kullanımından yüksektir. Sonuçlar simülasyonla üretildiği için simülasyon

parametrelerinin değiştirilmesi, sonuçların nümerik değerlerini etkileyebilir, ancak sonuçların anlamını değiştirmez.

### 3.6.2 Deney protokolü özeti ve metrik okuma

Her senaryo için deney akışı aynıdır: (1) “ScenarioConfig” veya görev havuzundan Görev üretimi. (2) beş iyileştirici algoritma + iki referans çizgisi koşusu. (3) Test grubu sonuçlarının grafik ve tablo arşivi. Gecikme ve kaynak kullanımı, Bölüm 3.4'e göre simütörden hesaplanmaktadır; mutlak ms değerleri görelî kıyas içindir (Mao vd. 2017).

Yerel ve uç referans çizgileri, optimizasyon yapılmadan referans sağlamaktadır (Bölüm 3.5.2). İyileştirici algoritmalar arasında en iyi kuralı, PSO/GWO keşif sırasında ortalama skorun düşmesini engeller (Kennedy vd. 1995). Gerçek dünya veri seti deneylerinde aynı kural ve aynı  $F(x)$  geçerlidir (Bölüm 3.5.3, 4.1.3); AS3 bu noktada test edilmektedir. On beş senaryonun tam  $\lambda$  listesi Ek B'de verilmiştir; özet merkez değerler Çizelge 3.7'dedir (Bölüm 3.1.1).

Karar verici algoritmaların karşılaştırılması da zorlu bir iştir. Algoritmaların popülasyonları yeni çözümler denediği için yalnızca en iyi performans gösteren bireylerin karşılaştırılması uygun olacaktır. Bu kararın nedeni, karar algoritmalarındaki; özellikle PSO ve GWO için, popülasyon üyelerinin keşif yapmasının, algoritmaların ortalama sonuçlarını düşürmesidir. GA elitleri için PSO ve ACO'nun en üst %20'si, GWO için alfa kurtlar ve ABC için elitler dikkate alınmaktadır. Bu yaklaşımla her algoritmanın en iyi performansları birbiriyle karşılaştırılmaktadır. Algoritmalar arasındaki karşılaştırma, güvenilir bir karşılaştırma yapmak için yeterlidir. Bu nedenle, referans çizgisi sonuçları da sonuca eklenmiştir. İki referans çizgisi koşulu eklenmiştir: Tüm yerel yürütme ve tüm uç nokta yürütmesi. Adlarından da anlaşılacağı üzere, bunlar tüm görevlerin yerel veya uç nokta sunucusunda çalıştırıldığı durumları temsil etmektedir.

### 3.6.3 Gerçek dünya veri setleri ve veri seti deney modülleri

Poisson tabanlı senaryo üreticine ek olarak, bu alt bölümde beş gerçek dünya veri seti

üzerinde yürütülen deneysel modüller tanımlanmaktadır. Modüller, veri seti adaptörü, kalibrasyon, hibrit birleşim üzerinde aynı beş meta-sezgisel algoritmayı (GA, PSO, GWO, ACO, ABC) çalıştırmaktadır. Böylece sentetik senaryo eksenini ile harici veri kökenli eksen, ortak standart öznitelik şeması ve ortak  $F(x)$  altında karşılaştırılabilir hale getirilmektedir (Bölüm 3.5. Bölüm 4.1.3).

Gerçek dünya veri setleri bu çalışma kapsamında tanımlanmış otuz iki MEC alanının tamamını ve YEREL/UÇ yerleşim etiketini birlikte sağlamamaktadır (Mohammadi vd. 2018). Alibaba ve Azure veri setleri öncelikle CPU, bellek ve görev süresi bilgisi vermektedir. GeoLife GPS mobilitesini; 5G Monarch saha ölçümü kanal koşullarını. MLPerf/CNN-DailyMail ise çıkarım yükü boyutunu temsil etmektedir. Bu eksiklik, üç katmanlı tamamlama ile giderilmektedir: (a) Poisson üretici eksik boyutları kontrollü biçimde doldurmaktadır. (b) “DataAdapter” veri seti sütunlarını standart öznitelik şemasına eşlemektedir. (c) “DataCalibrator” min-max ölçekleme ile değerleri simülasyon TARGET\_BOUNDS aralığına hizalamaktadır (Guo vd. 2019) (Hadary vd. 2020).

Bu çalışmada kullanılan beş kaynak aşağıda özetlenmektedir: ALIBABA\_V2 (Alibaba cluster trace v2018) (Guo vd. 2019). AZURE\_PUBLIC (Azure VM packing trace) (Hadary vd. 2020). CRAWDAD/GeoLife (GPS mobilite) (Zheng vd. 2010). 5G\_MONARCH (INFOCOM 2023 saha grid ölçümü) (Liu vd. 2023). MLPERF\_EDGE (uç çıkarım yükü) (Hermann vd. 2015) (Reddi vd. 2020). Her kaynak, standart öznitelik vektörünün yalnızca bir alt kümesini doğrudan karşılamaktadır. Kalan alanlar adaptör ve üretici katmanlarıyla tamamlanmaktadır.

Veri setinden görev örneği üretimi, trace tabanlı çalışmalardaki örnek (instance) oluşturma yaklaşımı ile uyumlu tutulmaktadır (Guo vd. 2019) (Hadary vd. 2020). Gecikme ve kaynak metrikleri ise Mao vd. (2017) ve Liu vd. (2020) ile kavramsal olarak örtüşmektedir. Tüm harici koşullarda protokol sabit tutulmaktadır: otuz iki standart sütun, aynı beş algoritma, aynı  $F(x)$  ve yalnızca gerçek veri satırlarından örnekleme zorunluluğu (require\_real\_data). Bu tutarlılık, Poisson senaryolarındaki on beş profilli sistematik kapsamı tamamlayan ikinci bir deneysel eksen oluşturmaktadır. Üç deney modu aşağıda özetlenmektedir. Dataset-only modunda tek veri seti havuzu kullanılmaktadır. Hibrit

modda `generate_task_basic()` ile veri seti kaydı ağırlıklı birleşim üretilmektedir. Adaptör+kalibrasyon modunda beş kaynak birleştirilmekte ve tek global kalibratör uygulanmaktadır.

Meta-sezgisel entegrasyon, optimizier sınıflarına eklenen “`task_factory`” parametresiyle sağlanmaktadır. Varsayılan davranış değiştirilmemiştir; Poisson senaryolarında `generate_task_for_scenario` kullanılmaya devam edilmektedir. Gerçek dünya veri seti deneylerinde havuzdan örneklenen görev için hesaplanan  $F(x)$ , sentetik senaryo koşullarındaki  $F(x)$  ile özdeş kabul edilmektedir. Böylece algoritmalar arası farklar, veri kökeninden değil arama operatörlerinden kaynaklanacak biçimde yorumlanabilmektedir (Bölüm 3.5.2).

#### **3.6.4 Otuz iki değişken ile gerçek dünya veri setlerinin uyumluluğu**

Bu alt bölümde, simülatörün beklediği otuz iki görev değişkeninin beş gerçek dünya veri setiyle ne ölçüde örtüştüğü açıklanmaktadır. Ham veri kayıtları doğrudan optimizasyona verilmemekte; önce adaptör katmanı ile standart öznitelik şemasına dönüştürülmektedir. Her değişken için üç durum sınıflandırılmaktadır: (i) doğrudan — veri setinde aynı veya çok yakın anlamlı sütun bulunmaktadır. (ii) türetilbilir — mevcut sütunlardan formül veya aralık eşlemesiyle üretilmektedir. (iii) yok — karşılık bulunmamakta, Poisson üretici veya temel varsayılanlar ile tamamlanmaktadır.

Otuz iki değişken altı grupta toplanmaktadır: coğrafi (mesafe, arazi, hava, kullanıcı yoğunluğu vb.), görev (boyut, karmaşıklık, tekrar, gecikme toleransı), yerel istemci kaynakları (CPU, bellek, depolama, ağ kartı, veriyolu), sunucu (tip, CPU/bellek/ağ kullanılabilirliği, kuyruk, mobilite), ağ (bağlantı tipi, hız, güvenilirlik, düğüm sayısı) ve dış ortam (kullanıcı hızı/yönü, kalabalık etkisi). Bu gruplama Bölüm 3.1.2'deki Standart öznitelik şeması ile uyumludur ve hangi veri setinin hangi boyutu taşıdığının okunmasını kolaylaştırmaktadır.

Alibaba Cluster Trace v2018 (Guo vd. 2019) ve Azure VM packing trace (Hadary vd.

2020) öncelikle veri merkezi iş yükünü temsil etmektedir. Görev boyutu ile CPU, bellek ve sunucu kapasite alanları doğrudan veya türetilbilir biçimde karşılanmaktadır. Buna karşılık coğrafi konum, mobilite ve birçok ağ alanı zayıftır; yaklaşık on beş alan üreteç veya adaptör ile tamamlanmaktadır. İki veri seti de benzer bir kısmi uyum profili göstermektedir.

GeoLife / CRAWDAD mobilite kaydı (Zheng vd. 2010) mesafe, kullanıcı hızı, yön ve kalabalık etkisini belirgin biçimde yansıtmaktadır. Görev boyutu ve sunucu kaynak alanları ise doğrudan bulunmamaktadır; bu veri setinde yaklaşık yirmi iki alan eksik kalmakta ve tamamlama katmanına ihtiyaç duyulmaktadır. Mobilite ağırlıklı veri setleri, dış çevre ve coğrafi boyutun deneysel eksende nasıl temsil edilebileceğini göstermektedir.

5G saha ölçümü INFOCOM 2023 (Liu vd. 2023) bağlantı hızı ve güvenilirliği açısından güçlüdür. Görev boyutu zayıf kalmaktadır. MLPerf / CNN-DailyMail (Hermann vd. 2015) (Reddi vd. 2020) ise çıkarım yükü ve görev karmaşıklığını taşımaktadır. Coğrafi ve mobilite alanları bulunmamaktadır. Bu nedenle 5G ve MLPerf kaynakları birbirini tamamlayıcı profiller sunmaktadır: biri ağ koşullarını, diğeri iş yükü boyutunu güçlendirmektedir. Çizelge 3.11'de değişken kategorilerinin veri seti eksenindeki kapsamı özet biçimde sunulmaktadır. Okuma kuralı şöyledir: “Doğrudan” sütunda alanın veri setinde karşılığı bulunmaktadır; “Türetilbilir” sütunda alan adaptör veya formül ile üretilmektedir; “Yok” sütununda ise üreteç veya varsayılan ile doldurulmaktadır.

Çizelge 3.11 Veri seti kategorisi × değişken kategorisi (özet)

| Değişken kategorisi | MLPerf                  | 5G ölçüm                | GeoLife                 | Alibaba/Azure           |
|---------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| Coğrafi             | Yok                     | Türetilbilir            | Türetilbilir            | Türetilbilir (bölge)    |
| Görev               | Doğrudan                | Türetilbilir            | Yok                     | Doğrudan / Türetilbilir |
| Yerel               | Doğrudan / Türetilbilir | Türetilbilir            | Yok                     | Türetilbilir            |
| Sunucu              | Doğrudan / Türetilbilir | Türetilbilir            | Yok                     | Doğrudan / Türetilbilir |
| Ağ                  | Türetilbilir            | Doğrudan / Türetilbilir | Türetilbilir            | Türetilbilir            |
| Dış ortam           | Yok                     | Türetilbilir            | Doğrudan / Türetilbilir | Yok                     |

Genel olarak gerçekçi bir MEC örnek vektörü için hibrit tamamlama önerilmektedir: görev ve kaynak boyutları Alibaba veya Azure veri setlerinden, mobilite ve dış ortam GeoLife'tan alınmakta; eksik alanlar Poisson üreticiden veya temel varsayılandan tamamlanmaktadır (Shakarami vd. 2020) (Mach vd. 2017). Beş gerçek dünya veri seti için değişken uyumluluk tabloları Ek C'de verilmiştir (Bölüm 3.5.4).

### 3.6.5 Literatürde veri seti kullanımı ve bağlam uyumu

Bu alt bölümde, beş gerçek dünya veri setinin literatürdeki tipik kullanım biçimi özetlenmekte ve bu tezdeki deneysel bağlamla nasıl hizalandığı açıklanmaktadır. Temel ilke şudur: veri seti, ham kayıt olarak değil; MEC aktarım bağlamına uygun görev örneği üreten bir kaynak olarak ele alınmaktadır (Guo vd. 2019) (Hadary vd. 2020) (Zheng vd. 2010).

Alibaba ve Azure veri setleri literatürde veri merkezi iş yükü ve kapasite baskısını temsil etmek için kullanılmaktadır. Alibaba çalışmalarında üretim benzeri CPU/bellek dağılımları ve kaynak çekişmesi korunarak zaman çizelgeleme ve yerleştirme örnekleri üretilmektedir (Guo vd. 2019). Azure veri setinde ise VM istek ve ömür dağılımlarıyla büyük ölçekli paketleme değerlendirmesi yapılmaktadır (Hadary vd. 2020). Bu tezde her iki kaynak da görev boyutu ve sunucu kaynak alanlarını güçlendirmekte; coğrafi ve mobilite boyutları tamamlama katmanıyla desteklenmektedir.

GeoLife GPS yörünge kaydı kullanıcı hareketi ve yörünge dinamiğini taşımaktadır. Literatürde mobilite ve konum tabanlı senaryolar için kullanılmaktadır (Zheng vd. 2010). 5G saha ölçümü bağlantı kalitesi ve çıktı (throughput) değişkenliğini modellemekte. Kanal koşullu uç kararları için referans oluşturmaktadır (Liu vd. 2023). MLPerf ve CNN/DailyMail ise standart çıkarım yükü bağlamında görev boyutu ve karmaşıklık profili sağlamaktadır (Reddi vd. 2020) (Hermann vd. 2015). Böylece her veri seti farklı bir MEC boyutunu güçlendirmekte, tek başına tam kapsam sunmamaktadır.

Dataset-only deneylerinde literatürle tutarlılık için beş varsayılan sabit tutulmaktadır.

“require\_real\_data=True” ile gerek veri iddiası varken sentetik dūřuř engellenmektedir. “use\_minmax\_calibration=False” ile veri seti bařına yeniden lekleme yapılmamakta, goreli veri seti leęi korunmaktadır. “drop\_missing\_rows=True” ile eksik satırlar gizlice doldurulmak yerine elenmektedir ve seffaflık artırılmaktadır. “execution\_sample\_cap=None” ile sabit satır tavanı zorlanmamaktadır. Alt kme kullanıcı seimine bırakılmaktadır. “clamp\_to\_simulator\_bounds=True” ile yalnızca simlatrn kabul etmedięi u deęerler kırpılmaktadır. Ham veri anlamı simlasyona hazırlık ařamasında mmkn olduęunca korunmaktadır.

Deęerlendirme ve raporlama tm meta-sezgisellerde ortak protokolle standartlařtırılmaktadır. Her aday Grev iin nce yerel, ardından u yolu simlatrde hesaplanmaktadır. Daha dřk  $F(x)$  veren yerleřim seilmektedir. Sonu ıktılarında poplasyon ortalaması deęil, arama sonundaki en iyi bireyin tek deęerlendirmesi raporlanmaktadır. Birincil stnlar iyileřtirici yanıt sresi (ms), yerleřim (yerel/u), gecikme, yrtme sresi ve veri seti etiketleridir. Kaynak Kullanımı stnunu bilinli olarak ayrı raporlanmamaktadır. nk  $F(x)$  iinde kaynak zaten kodlanmaktadır. Ham veri setinde ise u kapasite etiketi bulunmamaktadır (Guo vd. 2019) (Mao vd. 2017). Poisson senaryo deneylerinde gecikme ve kaynak birlikte raporlanmaya devam edilmektedir (Mach vd. 2017).

Zamansal dilimleme  $\text{temporal\_part} \in \{\text{all}, \text{head}, \text{tail}\}$  ve  $\text{temporal\_ratio}$  ile uygulanmaktadır; zaman damgası stnunu yoksa satır sırası tabanlı blme kullanılmaktadır. Tekrarlı kořularda hibrit ve adaptr modları iin “run \*\_repeated” API'leri ortalama $\pm$ std retmektedir (Blm 4.1.9).

### 3.6.6  deney modunun ayrıntılı metodolojisi

Gerek dnya veri seti deneyleri, Poisson senaryo dngsnden baęımsız  modlde yrtlmektedir. Tm modlarda GA, PSO, GWO, ACO ve ABC ile aynı  $F(x)$  kullanılmaktadır; bylece yalnızca grev havuzu retim yntemi deęiřtirilmekte, algoritma karřılařtırması adil tutulmaktadır (Mao vd. 2017). Modlar arasındaki farklar izelge 3.12'de zetlenmektedir.

**Mod A (Dataset-only):** Tek bir veri seti yüklenmekte, isteğe bağlı zamansal dilim uygulanmakta ve eksik satırlar elenmektedir. Veri seti satırı DataAdapter ile standart öznitelik şemasına dönüştürülmekte; simülatör sınırları dışındaki değerler TARGET\_BOUNDS ile kırılmaktadır. Ardından her görev için yerel ve uç yolları değerlendirilmekte ve beş algoritma aynı havuz üzerinde çalıştırılmaktadır. Bu mod, her veri setinin kendi profilinde bağımsız kıyas üretmek için temel referans olarak kullanılmaktadır.

**Mod B (Hibrit):** Sentetik “generate\_task\_basic()” ile üretilmekte; veri seti kaydı özellikleri  $\alpha$  ağırlığıyla birleştirilmektedir (varsayılan  $\alpha=0,62$ ). Poisson senaryo döngüsüne karıştırılmamakta; eksik veri seti boyutlarının kontrollü tamamlanması amaçlanmaktadır (Bölüm 3.5.6).

**Mod C (Adaptör + kalibrasyon, birleşik):** Beş kaynak “joined\_data\_utils” ile birleştirilmekte; tek global “DataCalibrator” ve “DataAdapter” uygulanmaktadır. Hibrit karışım ve kümeleme kullanılmamaktadır. Bu mod, farklı kaynakların aynı ölçekte tek havuzda toplanmasını sağlamaktadır.

Ortak API “research\_api.py” ve orkestrasyon “research\_pipeline.py” üzerinden sağlanmaktadır. Poisson tabanlı konfigürasyon döngüsü değiştirilmemiştir; harici veri modları yalnızca “task\_factory” parametresi ile optimizör sınıflarına bağlanmaktadır. Böylece sentetik ve gerçek dünya deneyleri aynı değerlendirme implementasyonunu paylaşmaktadır.

Çizelge 3.12 Deney modları karşılaştırması

| Mod               | Veri kaynağı                 | Sentetik karışım  | Kalibrasyon   | Tipik kullanım         |
|-------------------|------------------------------|-------------------|---------------|------------------------|
| Poisson senaryosu | Poisson $\lambda$ profilleri | Hayır             | Hayır         | 15 sistematik senaryo  |
| Dataset-only      | Tek veri seti                | Hayır             | Hayır (vars.) | Veri seti başına kıyas |
| Hibrit            | Kayıt + sentetik             | Evet ( $\alpha$ ) | İsteğe bağlı  | Eksik boyut tamamlama  |
| Adaptör+birl eşik | Beş kayıt birleşik           | Hayır             | Evet (global) | Kalibre birleşik havuz |

### 3.6.7 Metodolojik sınırlamalar ve gelecek iyileştirmeler

Bölüm 3.5.3'da tanımlanan deneysel çerçeve, Poisson tabanlı on beş senaryo ile gerçek dünya veri seti eksenini aynı  $F(x)$  ve aynı meta-sezgisel protokol altında birleştirmeyi amaçlamaktadır. Bu bütünlük, tezin AS1–AS4 araştırma sorularına yanıt verebilmesi için gerekli olmakla birlikte, veri dönüşümü, istatistiksel raporlama ve simülatör güvenliği katmanlarında bilinçli mühendislik tercihleri içermektedir. Aşağıdaki alt başlıklarda bu tercihlerin sınırları, bulguların yorumlanmasına etkileri ve gelecek çalışmalara yönelik iyileştirme önerileri sistematik biçimde açılmaktadır.

#### 3.6.7.1 Zamansal bölme ve zaman damgası bağımlılığı

Gerçek dünya veri seti deneylerinde zamansal dilimleme, “temporal\_part”  $\in \{all, head, tail\}$  parametresi ve “temporal\_ratio” eşiği ile uygulanmaktadır (Bölüm 3.5.6). Mevcut uygulamada, kayıta ayrı bir zaman damgası sütunu bulunmadığı durumlarda bölme işlemi satır sırasına dayanmaktadır: veri setinin ilk %r oranı “head”, son %r oranı “tail” dilimi olarak ayrılmakta; “all” seçeneği tüm kayıtları kapsamaktadır. Bu yaklaşım, sonuç kayıtlarının çoğunda pratik bir çözüm sunmakta ve tekrarlanabilirliği kolaylaştırmaktadır.

Bununla birlikte, satır sırası tabanlı bölmenin metodolojik sınırı açıktır. Alibaba ve Azure veri setlerinde satırların kronolojik sırayla dizilmiş olduğu varsayımı her zaman doğrulanmamaktadır; GeoLife gibi mobilite kayıtlarında ise zaman bilgisi ayrı bir alanda bulunmakta, satır sırası ile gerçek zaman arasında doğrudan eşleşme kurulamamaktadır. Bu durum, head ve tail dilimlerinin «erken dönem» ve «geç dönem» trafiğini temsil ettiği yorumunu zayıflatmaktadır. Guo vd. (2019), gerçek kayıt tabanlı veri setlerden görev örneği üretirken zaman penceresinin açık biçimde tanımlanmasının dış geçerlilik için kritik olduğunu vurgulamaktadır.

Gelecek iyileştirme önerileri şunlardır: (i) kayıta timestamp veya eşdeğer bir sütun varsa timestamp-aware bölme; (ii) zaman damgası olmayan veri setlerinde bölme stratejisinin raporda açıkça belirtilmesi; (iii) head–tail karşılaştırmalarının yalnızca zamansal

değişkenlik iddiası taşıyan çalışma izlerinde kullanılması; mobilite ve çıkarım yükü veri setlerinde alternatif bölme kriterlerinin (coğrafi bölge, görev boyutu) değerlendirilmesi.

### **3.6.7.2 İstatistiksel testler ve çoklu karşılaştırma raporlaması**

Meta-sezgisel arama doğası gereği stokastiktir: aynı görev havuzu ve aynı  $F(x)$  altında bile başlangıç tohumu (seed) değişikçe en iyi birey ve algoritma sıralaması farklılaşabilmektedir (Goldberg 1989; Bölüm 3.5). Bu belirsizliği azaltmak için dataset-only ve birleşik modlarda  $n = 10$  tekrarlı koşu protokolü uygulanmış; sonuçlar ortalama  $\pm$  standart sapma biçiminde Çizelge 4.7 ve Çizelge 4.9'da özetlenmiştir (Bölüm 4.1.9). Poisson senaryo ekseninde ise Bölüm 4.1'deki bulgular çoğunlukla tek koşuluk ölçümlere dayanmakta; tekrarlı değerlendirme bu ekseninde öneri düzeyinde bırakılmıştır (Bölüm 4.1.10).

Mevcut raporlama düzeyinin sınırı, tanımlayıcı istatistiklerin (ortalama, std) inferans istatistiklerine (hipotez testi, güven aralığı, etki büyüklüğü) tam olarak dönüştürülmemiş olmasıdır. İki algoritmanın aynı veri setinde gerçekten farklı performans gösterip göstermediği, Wilcoxon işaretli sıra testi ile sınımaaya uygundur; çünkü test, eşleştirilmiş tekrar çıktılarını kullanmakta ve normallik varsayımı gerektirmemektedir (Wilcoxon 1945). Beş algoritma ve iki referans çizgisi arasında çoklu ikili karşılaştırma yapıldığında ise aile-içi hata oranının (FWER) kontrolü için Bonferroni veya Benjamini–Hochberg düzeltmesi gerekir. Bu tezde Wilcoxon ve Bonferroni'nin otomatik hesaplanıp tabloya aktarılması gelecek iş olarak planlanmıştır (Bölüm 4.2.6).

### **3.6.7.3 Simülatör sınır kırpması ve ölçek patlaması riski**

Dataset-only modunda beş varsayılan sabit tutulmaktadır; bunlardan biri `clamp_to_simulator_bounds=True` (B seçeneği) ile simülatörün kabul ettiği `TARGET_BOUNDS` aralığına sayısal kırpmaya uygulanmasıdır (Bölüm 3.5.4; Bölüm 3.5.6). Kırpmaya, ceza fonksiyonu yerine tercih edilmiştir: sınır dışı adaylar elenmez, sınırdaki tutulur; böylece simülatör istikrarı korunur ve arama uzayı kesintisiz kalır (Hadary

vd. 2020; Ben-David vd. 2010).

Bu tercihin pratik gerekçesi, Azure VM packing (AZURE\_PUBLIC) veri setinde gözlemlenen ölçek patlaması olayıdır. `clamp_to_simulator_bounds=False` iken yapılan erken tekli koşuda GA, simülatör ölçeğinin çok üzerinde gecikme değerleri üretebilmiş; bu durum hem algoritma sıralamasını çarpıtmakta hem de karşılaştırmanın adillğini zayıflatmaktadır. Kırpma etkinleştirildiğinde tüm algoritmalar 12–17 ms bandına sıkışmakta; PSO en düşük yanıt süresini elde etmektedir (Bölüm 4.1.3). Dar bant, güvenli arama uzayına hizalamanın önemini göstermektedir.

Metodolojik sınır şudur: `use_minmax_calibration=False` ile veri seti başına min–max yeniden ölçekleme kapalı tutulmaktadır. Bu, veri setleri arası görelî ölçeğî korumak için bilinçli bir tercihtir; ancak ham kayıttaki aşırı uç değerler kırpma ile sıkıştırıldığında algoritmalar arası farklar da daralmaktadır. Dolayısıyla Azure sonuçları, «algoritma üstünlüğü»nden çok «güvenli arama uzayında görelî sıralama» olarak yorumlanmalıdır.

Gelecek iyileştirme önerileri şunlardır: (i) kırpma öncesi/sonrası değer dağılımının (histogram veya yüzdelik dilim) raporlanması; (ii) iki katmanlı kısıt stratejisinin (Kısıt-1: adaptör kırpması; Kısıt-2:  $F(x)$  içi  $\max(\cdot, 1)$  normalizasyonu) her veri seti için ayrı ayrı belgelenmesi; (iii) kalibrasyon açık/kapalı ve kırpma açık/kapalı için  $2 \times 2$  ablation matrisi; (iv) ölçek patlaması riski taşıyan veri setlerinde arama uzayının veri setine özgü dinamik sınırlarla genişletilmesi.

#### **3.6.7.4 Gerçek veri zorunluluğu ve vekil (Proxy) veri politikası**

Veri seti deneylerinde “`require_real_data`” değişkeni, gerçek kayıt kullanım iddiasının denetim mekanizması olarak tanımlanmıştır. “`require_real_data=True`” iken, gerçek veri yüklenemezse deney sentetik moda düşmez; hata veya açık uyarı üretilerek şeffaflık hedeflenmiştir (Bölüm 3.5.6). Buna karşılık “`require_real_data=False`” olduğunda deterministik vekil (proxy) veri devreye girmektedir. Bu mod geliştirme, birim testi ve deney tutarlığı doğrulaması için uygundur, ancak kayıt-temelli bilimsel iddialar için

sınırlı kabul edilmelidir. Metodolojik sınır, vekil verinin gerçek kayıtla aynı dağılım özelliklerini taşımasıdır. Proxy modda üretilen örnek vektörleri, Poisson üretici veya sabit şablondan türetilmekte; Alibaba/Azure/GeoLife profillerindeki çarpık dağılım, kuyruk dinamiği veya mobilite örüntüsü yansıtılmamaktadır. Bu nedenle tezdeki tüm «gerçek dünya» bulguları, `require_real_data=True` koşulu altında yürütülmüştür; geliştirme ortamında vekil kullanımı sonuç bölümüne dahil edilmemiştir.

### 3.6.8 Tekrarlanabilirlik, ortam ve yazılım sürümleri

Deneysel tekrarlanabilirlik için: (1) Rastgele değer (seed) kaydı; (2) Çoklu koşu ( $n \geq 10$ ) ve ortalama $\pm$ standart sapma raporu; (3) Koşu parametrelerinin ve değerlerin kayıt altına alınması. Deterministik bileşen: aynı görev  $x$  ve aynı simülatör girdisi  $\rightarrow$  aynı  $F(x)$ ; belirsizlik popülasyon başlangıcı ve seçim operatörlerindedir (Goldberg 1989).

İstemci Python, sunucu Java Spring Boot; HTTP görev iletimi kullanılmaktadır. Gerçek veri seti indirme “`download_real_datasets`” modülü ile yapılmaktadır.

### 3.6.9 Hesaplama maliyeti ve deneysel ortam

Deneysel ortam iki katmanlıdır: Python istemci optimizasyon döngüsünü yürütmekte, Java Spring Boot tabanlı MEC simülatörü ise her Görev için gecikme ve kaynak hesaplamaktadır. İstemci ile simülatör HTTP üzerinden (`localhost:8080`) haberleşmektedir. Koşular Windows 10 üzerindeki yerel geliştirme makinesinde gerçekleştirilmiştir. Bu mimari saha dağıtımını değil, kontrollü ve tekrarlanabilir karşılaştırma ortamını hedeflemektedir (Bölüm 3.5.6).

Hesaplama maliyeti iki ana bileşenden oluşmaktadır. Birinci bileşen meta-sezgisel aramadır; maliyet popülasyon büyüklüğü, iterasyon sayısı ve simülatör çağrı adediyle doğrusal biçimde artmaktadır (popülasyon=100, iterasyon=100). İkinci bileşen veri ön işlemedir; gerçek dünya veri setlerinin indirilmesi ve işlenmiş sonuç dosyalarının üretilmesi bu gruba girmektedir. Arama maliyeti tüm deneylerin baskın süresini

oluşturmaktadır.

Popölasyon veya iterasyon artırıldığında deęerlendirme sayısı ( $N_{eval}$ ) doğrusal olarak yükselmektedir. Bu çalışmada limitler sabit tutularak operatörler arası adil kıyas öncelenmiştir. Paralel simölator örnekleri veya dağıtık deęerlendirme gelecek çalışma kapsamında bırakılmıştır (Taleb vd. 2017).

## 4. BULGULAR VE TARTIŞMA

Karar algoritmaları tüm senaryolara uygulanarak sonuçlar elde edilmiştir. Bu sonuçlar bir sonraki kısımda paylaşılmıştır.

### 4.1 Test Sonuçları

Bu bölümde, Bölüm 3'te tanımlanan deneysel protokol kapsamında elde edilen Poisson senaryo ve gerçek dünya veri seti sonuçları sunulmaktadır. Öncelikle deneysel tasarım ve sonuç okuma ilkeleri özetlenmektedir. Ardından on beş senaryo için gecikme ve kaynak kullanımı bulguları Şekil 4.1– 4.36 ve ilgili çizelgelerle birlikte yorumlanmaktadır. Gerçek dünya veri seti eksen, tekrarlı koşu protokolü, birleşik modlar ve Poisson–kayıt çapraz kıyası izleyen alt başlıklarda açılmaktadır. Bölüm sonunda bulgular AS1–AS4 araştırma soruları ve literatürle ilişkilendirilerek sentezlenmektedir.

#### 4.1.1 Deneysel tasarım ve sonuç okuma ilkeleri

On beş senaryo, statik ve dinamik profilleri kapsayacak şekilde tasarlanmıştır (Bölüm 3.5). Her senaryoda beş optimizör ve iki referans çizgisi aynı batch ve popülasyon ayarlarıyla koşturulmuştur. Şekil 4.1–4.30 senaryo bazlı gecikme ve kaynak eğrilerini. Şekil 4.31–4.32 ortalama metrikleri. Şekil 4.33–4.36 dinamik/statik gruplamayı göstermektedir.

Yorumlama ilkesi: Mutlak milisaniye değerleri simülatör ölçeğindedir; senaryolar arası görelî sıralama esas alınmalıdır (Bölüm 3.4.2). Yüksek yoğunluklu kentsel, akıllı şehir, büyükşehir veri merkezi ve etkinlik trafiği senaryolarında PSO/GWO öne çıkması, agresif keşfin kaynak harcayarak gecikmeyi düşürmesi ile uyumludur (Mirjalili vd. 2014) (Kennedy vd. 1995). Sınırlı internet kırsal, tarımsal ağ ve düşük kaynak profillerinde GA/ABC üstünlüğü, elitizm ve koloni limit mekanizmaları ile açıklanır (Goldberg 1989) (Karaboga vd. 2007).

Dinamik senaryolarda tüm algoritmalarda performans düşüşü beklenir. GA'nın görece hızlı adaptasyonu popülasyon çeşitliliği ile ilişkilendirilmiştir. Referans çizgilerine göre tüm iyileştiricilerin iyileşme sağlaması,  $F(x)$  optimizasyonunun anlamlı olduğunu göstermektedir.

Gerçek dünya veri seti sonuçları (Bölüm 4.1.3) aynı ilkelerle okunmalı; tek kazanan algoritma olmaması, senaryo-şartlı seçim tezinin dış geçerliliğini desteklemektedir (Guo vd. 2019) (Hadary vd. 2020).

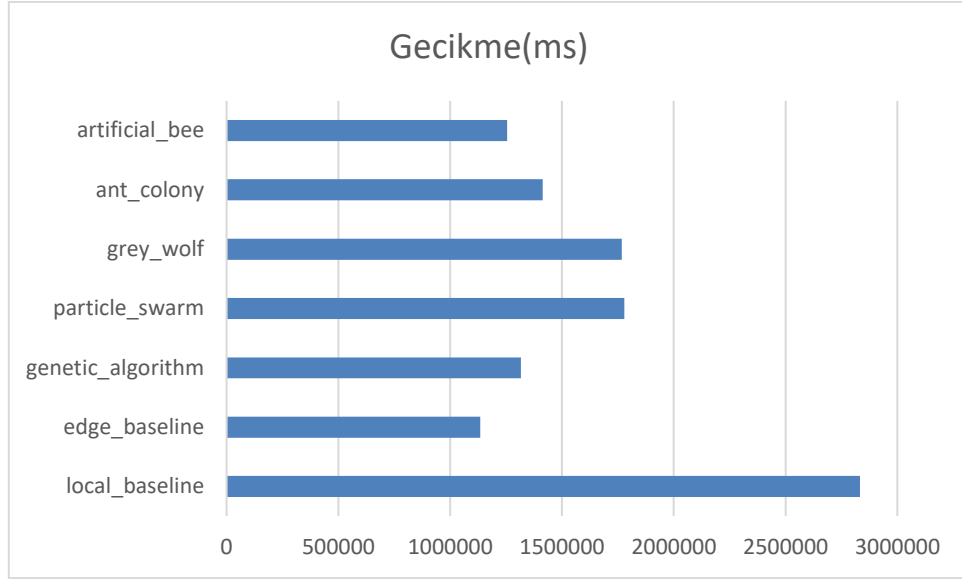
Her senaryo için başarı metriklerine; Gecikme (Delay) ve kaynak kullanımı (Resource Consumption); dair elde edilen sonuçlara aşağıda yer verilmiştir. Burada not edilmesi gereken bir nokta test sonuçlarının yorumlanmasına bir sonraki kısımda yer verileceğidir. Genel olarak kaynağın yüksek yoğunluklu senaryolarda GWO ve PSO öne çıkmaktadır. Düşük yoğunluklu senaryolarda GA ve ABC daha başarılı olmuştur. ACO ise diğer algoritmalara göre daha ortalama sonuçlar üretmiştir.

#### **4.1.2 Senaryo sonuçlarının yorumlanması**

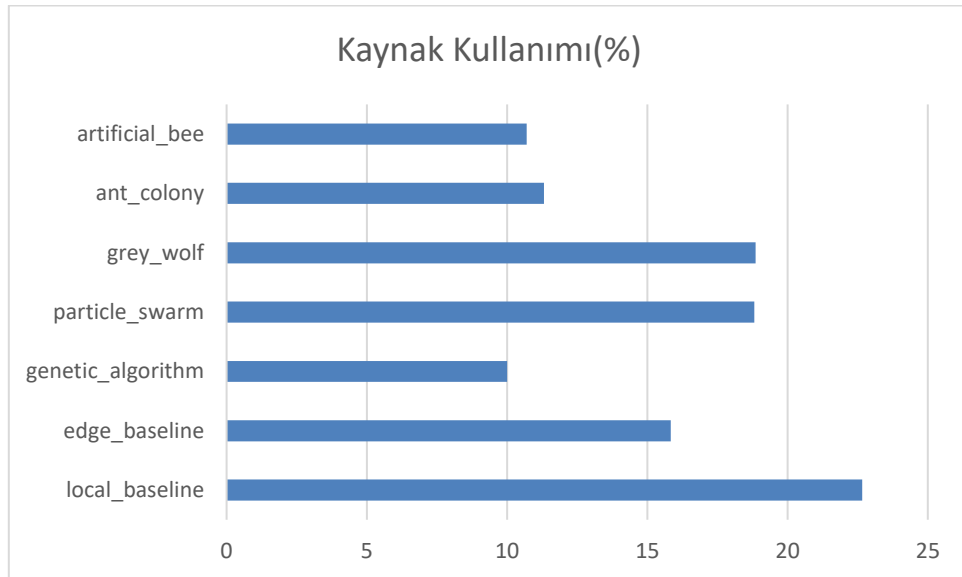
Bölüm 4.1'deki her senaryo paragrafı kısa özet niteliğindedir; izleyen yorum paragrafları literatür ve AS2 bağlantısını açar. Okuma sırası önerilir: (1) Şekil 4.x gecikme/kaynak eğrileri; (2) kısa özet paragraf; (3) senaryo yorumu. Genel örüntü AS1'i desteklemektedir: düşük kaynak ve orta yoğunluk profillerinde (profil A) GA/ABC, yüksek yoğunluk ve geniş bant profillerinde (profil B) PSO/GWO öne çıkmaktadır. ACO çoğu profilde orta sıradadır; orta kaynak koşulları feromon maliyetinin keşif kazancını sınırladığı için "tuzak bölge" olarak kalmaktadır (Dorigo vd. 2006). Referans çizgilerinin (tümü yerel / tümü uç) altına inen tüm iyileştiriciler,  $F(x)$  optimizasyonunun anlamlı olduğunu göstermektedir (Bölüm 3.5.2).

Temel Senaryo (Şekil 4.1–4.2): Çizelge 3.7'deki  $\lambda$  merkezi orta düzey kaynak ve yoğunluk profilini temsil etmektedir. Gecikme eğrilerinde GA ve ABC biraz daha düşük gecikme değerleri üretmiş, PSO ve GWO daha yüksek gecikme değerleri üretmiştir;

kaynak eğrilerinde GA daha düşük kaynak kullanımıyla dengeli bir profil oluşturmuştur. Tümü yerel ve tümü uç referanslarına göre iyileştiriciler ölçülebilir iyileşme göstermiştir. Bu profil, elitizm ve limit tabanlı aramanın (GA/ABC) dengeli koşullarda güvenilir olduğunu; agresif keşfin (PSO/GWO) henüz daha düşük bir bant oluşturmadığını göstermektedir (Goldberg 1989; Karaboga vd. 2007).



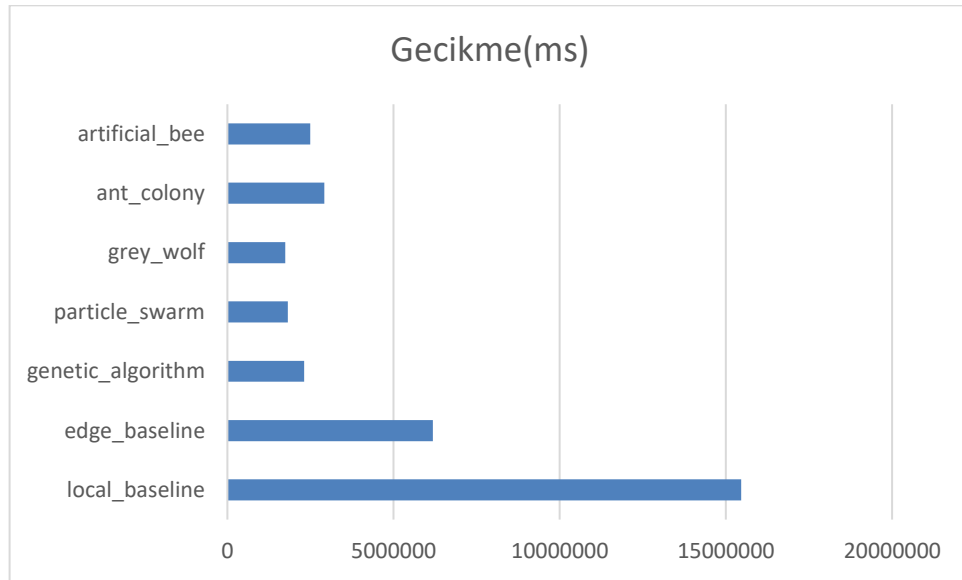
Şekil 4.1 Baz senaryo için gecikme



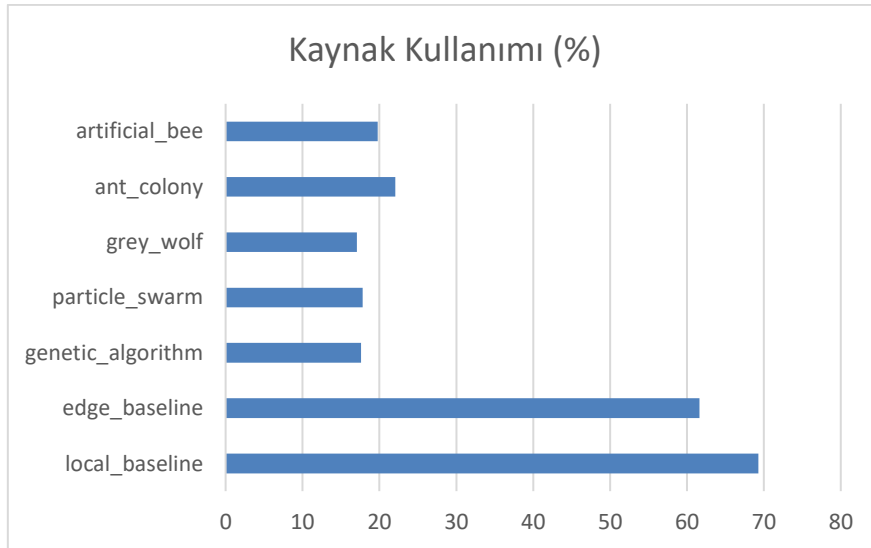
Şekil 4.2 Baz senaryo için kaynak kullanımı

Yüksek Yoğunluklu Kentsel Alan (Şekil 4.3–4.4): Yüksek  $\lambda$  ve yoğun görev varışı

profilinde GWO gecikme ve kaynak dengesinde en düşük değerleri üretmiş, PSO yakın performans vermiştir. GA yaklaşık %20 daha yüksek gecikme değerleri üretmiş; ABC ve ACO daha düşük sıradadır. Referans çizgilerine göre PSO/GWO bandı ölçülebilir iyileşme göstermiştir. Agresif keşif operatörleri yüksek yoğunlukta kaynak harcayarak gecikmeyi düşürmektedir; bu bulgu AS2 ile uyumludur (Kennedy vd. 1995; Mirjalili vd. 2014).

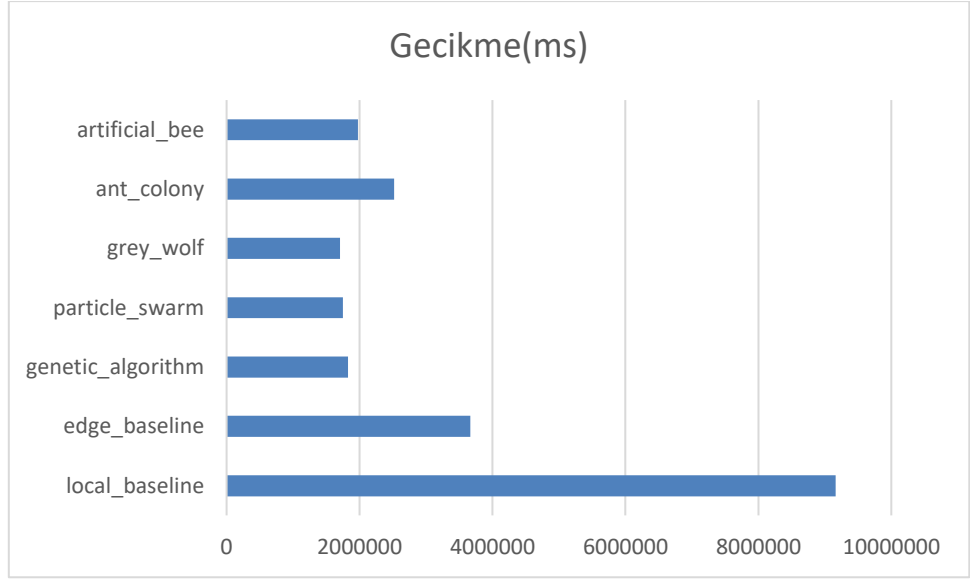


Şekil 4.3 Yüksek yoğunluklu kentsel alan için gecikme

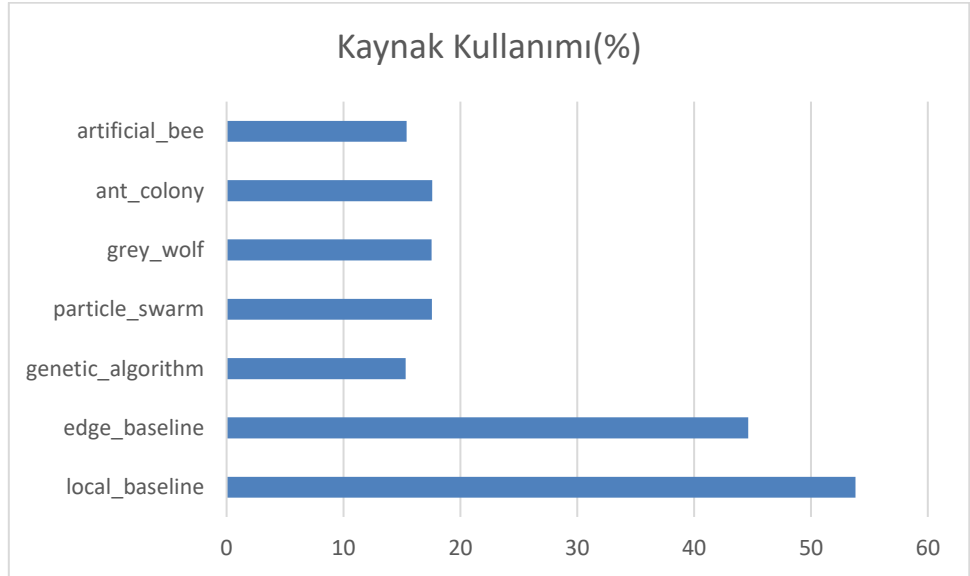


Şekil 4.4 Yüksek yoğunluklu kentsel alan için kaynak kullanımı

Akıllı Şehir Ağı (Şekil 4.5–4.6): IoT cihazlarının heterojen talep profilinde GWO ve PSO gecikmede en düşük değerleri üretmiştir; GA, ABC ve ACO yakın ancak daha yüksek gecikme değerleri üretmiştir. Kaynak kullanımında GA ve ABC küçük bir marjla daha verimli olmuştur. Dual-path değerlendirme, yoğun edge yükünde uç seçimine izin vermiştir (Mao vd. 2017). Yüksek bağlantı çeşitliliği profillerinde sürü tabanlı keşif, feromon tabanlı aramaya göre daha hızlı yakınsama eğilimi göstermiştir (Mach vd. 2017).

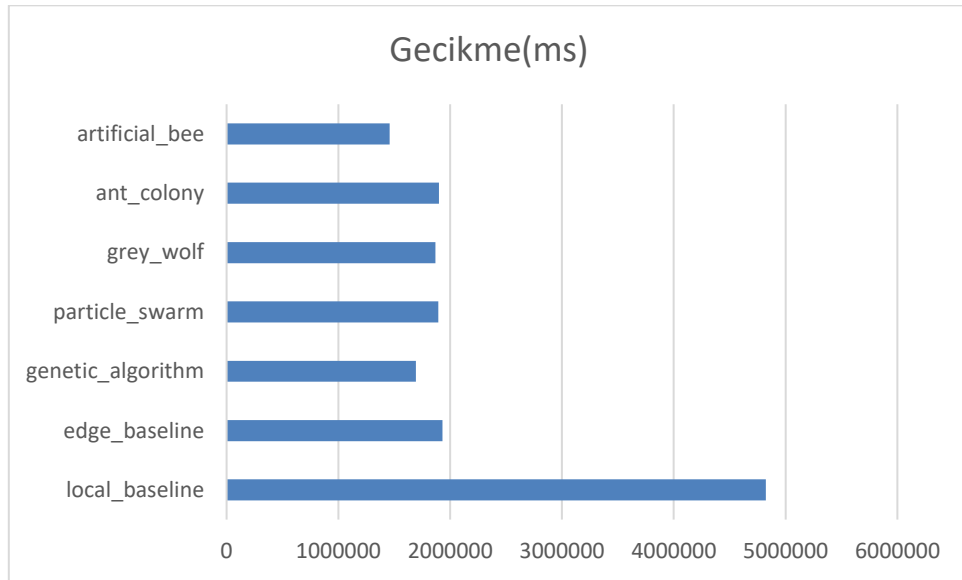


Şekil 4.5 Akıllı şehir ağı için gecikme

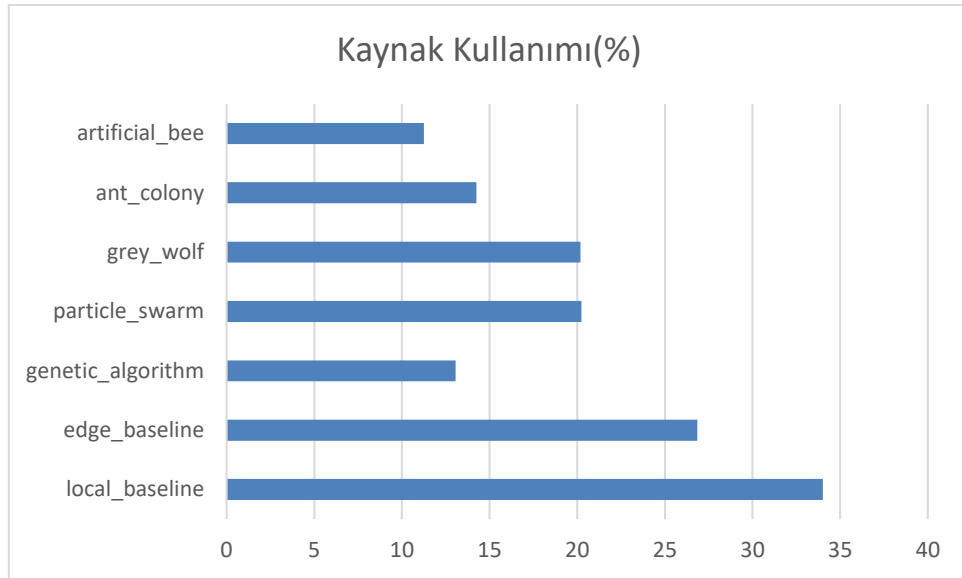


Şekil 4.6 Akıllı şehir ağı için kaynak kullanımı

Tüketici Ağı Kullanımı Sınırlı İş Bölgesi (Şekil 4.7–4.8): Orta-düşük kaynak ve sınırlı tüketici bant genişliği profilinde ABC iş yükü optimizasyonunda bu koşuda en düşük gözlemlenen gecikmeyi üretmiştir. PSO, GA ve GWO yakın ancak ABC'ye göre daha yavaş yakınsamıştır; ACO orta sıradadır. Kaynak kullanımında PSO ve GWO en yüksek tüketimi üretmiştir. Limit tabanlı ABC operatörü, kısıtlı profilde elitizmden daha dengeli davranmıştır (Karaboga vd. 2007).

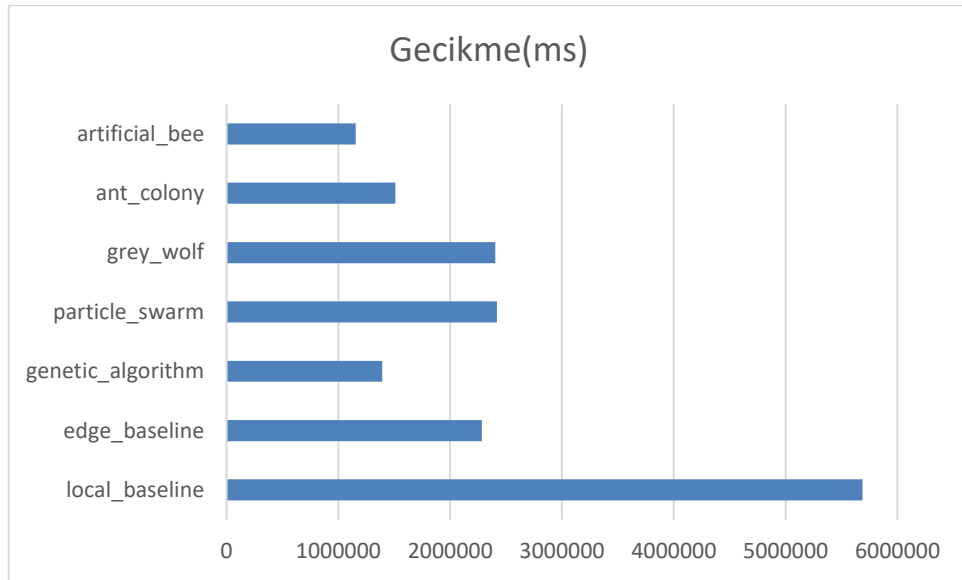


Şekil 4.7 Tüketici ağı kullanımı sınırlı iş bölgesi için gecikme

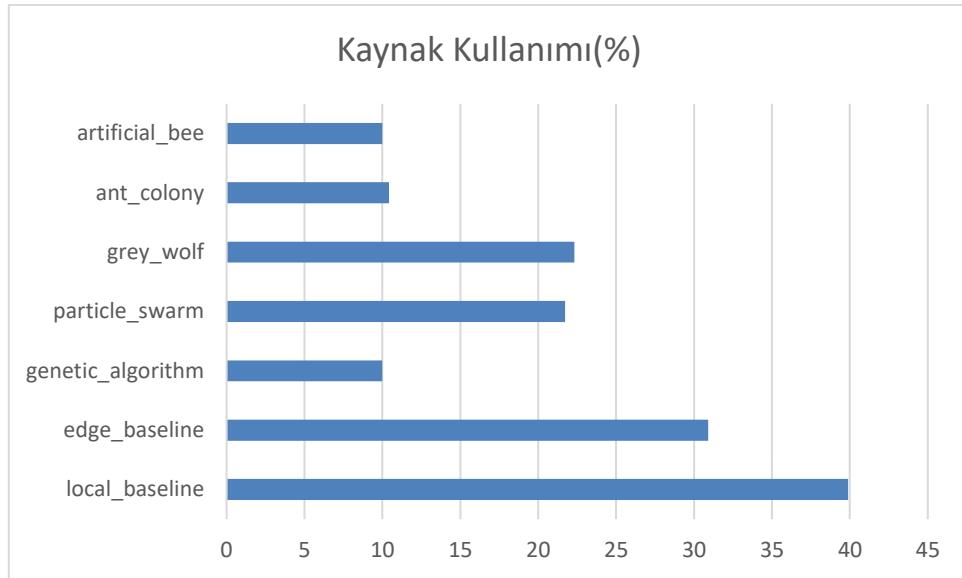


Şekil 4.8 Tüketici ağı kullanımı sınırlı iş bölgesi için kaynak kullanımı

Etkinlik Kaynaklı Yüksek Trafik (Şekil 4.9–4.10): Ani yük artışı ve dinamik trafik profilinde GA ve ABC gecikmede en düşük değerleri üretmiştir; PSO ve GWO yüksek yoğunlukta daha yüksek gecikme değerleri üretmiş, ACO orta bantta yer almıştır. Kaynak kullanımında ABC, ACO ve GA daha düşük kaynak kullanım değerleri gözlenmiştir. Popülasyon çeşitliliği, ani yük dalgalanmalarına adaptasyonu desteklemiştir (Goldberg 1989). Bu senaryo, düşük kaynak profillerinde AS2 ile uyumlu bulgular sunmaktadır.

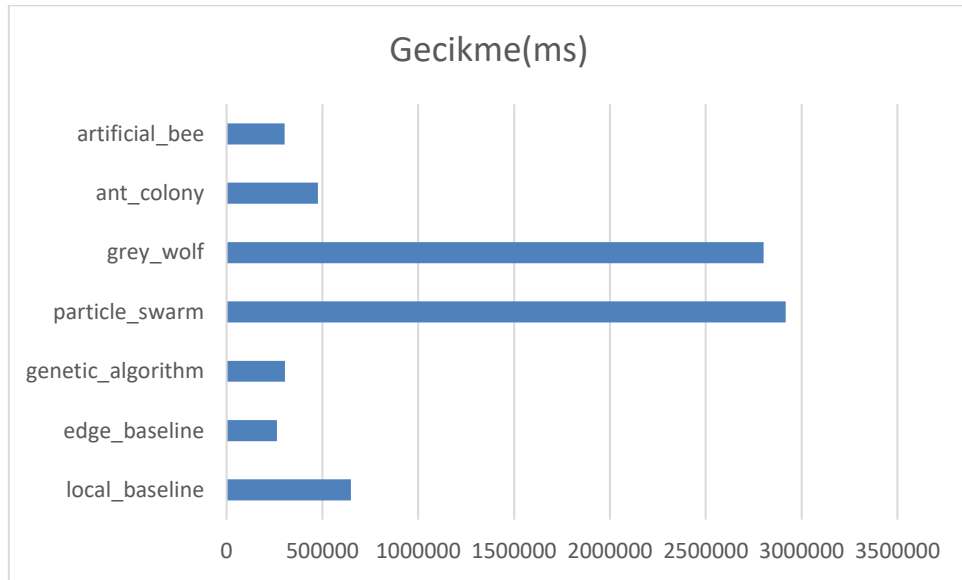


Şekil 4.9 Etkinlik kaynaklı yüksek trafik için gecikme

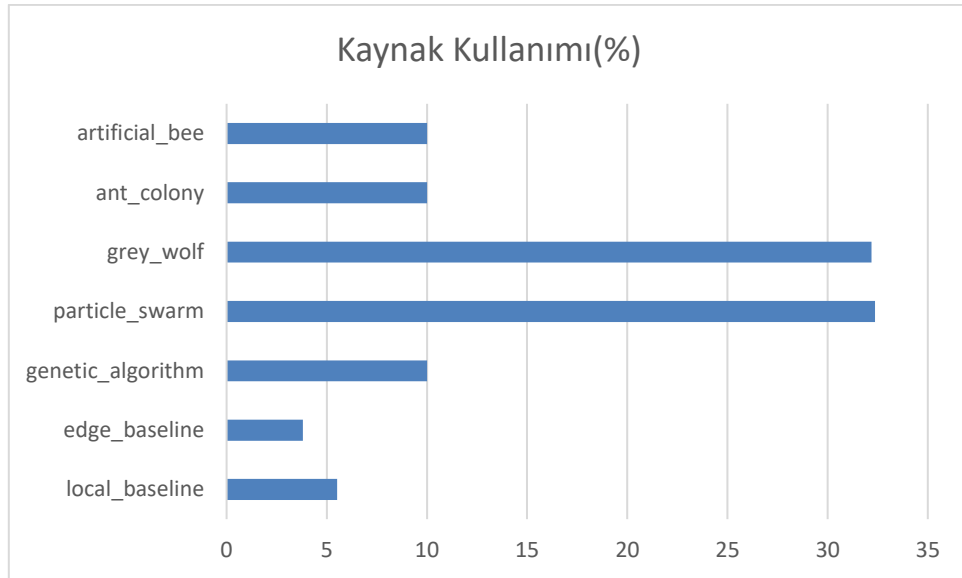


Şekil 4.10 Etkinlik kaynaklı yüksek trafik için kaynak kullanımı

Sınırlı İnternet Erişimi Olan Kırsal Bölge (Şekil 4.11–4.12): Düşük bant genişliği ve kısıtlı edge kaynağı profilinde GWO ve PSO daha yüksek gecikme değerleri üretmiştir. ABC, ACO ve GA daha düşük gecikme değerleri gözlenmiştir. Referans çizgilerine göre agresif keşif yapan algoritmaların riskli arama uzayı genişlemesi gecikmeyi artırmıştır. Düşük kaynak profillerinde limit ve feromon tabanlı arama görece avantaj göstermiştir (Dorigo vd. 2006; Karaboga vd. 2007).

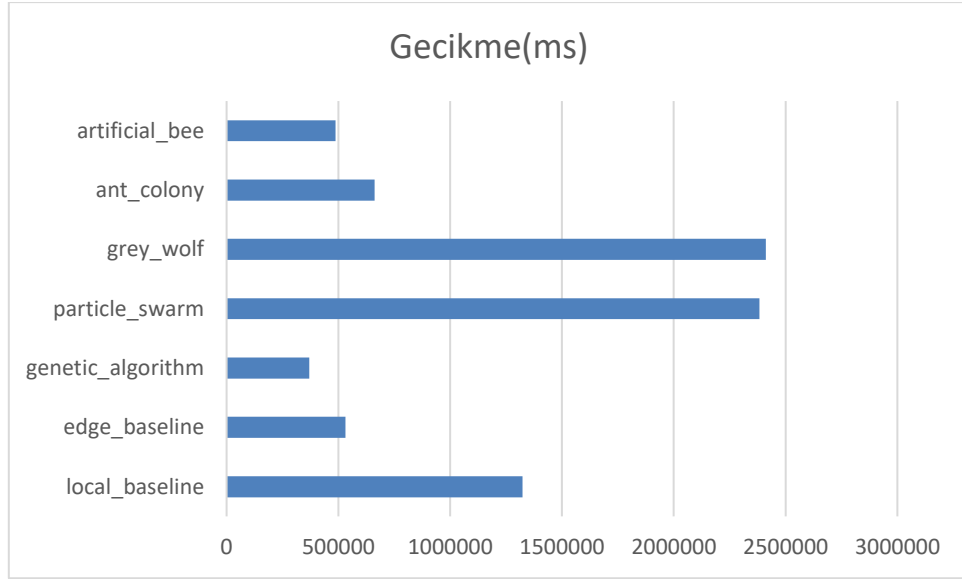


Şekil 4.11 Sınırlı internet erişimi olan kırsal bölge için gecikme

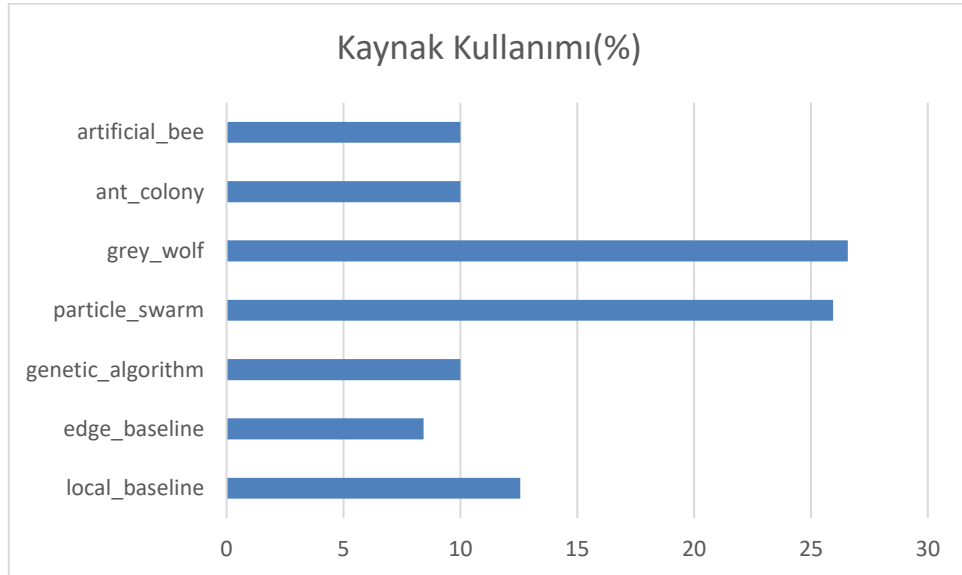


Şekil 4.12 Sınırlı internet erişimi olan kırsal bölge için kaynak kullanımı

Tarımsal Ağ İhtiyaçları (Şekil 4.13–4.14): Düşük kaynak ve seyrek görev varışı profilinde GA, ABC ve ACO beklendiği üzere daha düşük gecikme değerleri üretmiştir. GWO ve PSO birbirine yakın ancak daha yüksek gecikme değerleri üretmiştir. Kaynak eğrileri de aynı sıralamayı desteklemektedir. Kırsal düşük yoğunluk profillerinde elitizm ve limit tabanlı arama, agresif keşfe göre daha güvenilir sonuç üretmiştir (Goldberg 1989).



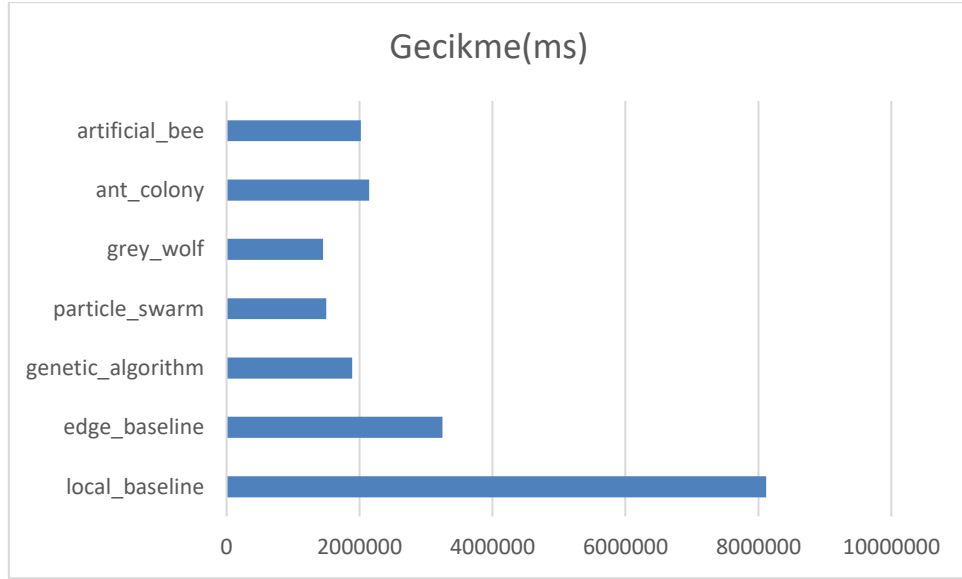
Şekil 4.13 Tarımsal ağ ihtiyaçları için kaynak kullanımı



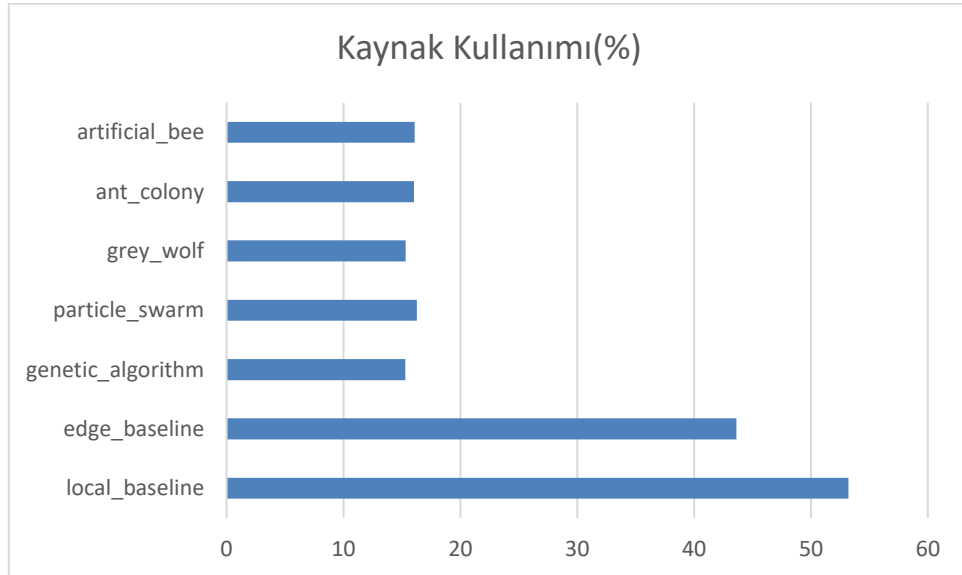
Şekil 4.14 Tarımsal ağ ihtiyaçları için kaynak kullanımı

Fiber Optik veya 5G Altyapısına Sahip Kırsal Bölge (Şekil 4.15–4.16): Yüksek bant

genişliği profilinde GWO ve PSO hafif önde olmakla birlikte tüm iyileştiriciler yüksek performans bandına yakınsamıştır. Algoritmalar arası fark azalmıştır; referans çizgilerine göre tüm aileler iyileşme sağlamıştır. Geniş bant, edge aktarımının cezasını düşürdüğünden keşif operatörleri görece avantajını kısmen kaybetmektedir (Liu vd. 2023). Bu profil, gerçek dünya 5G saha verisi bulgularıyla (Bölüm 4.1.3) tutarlıdır.



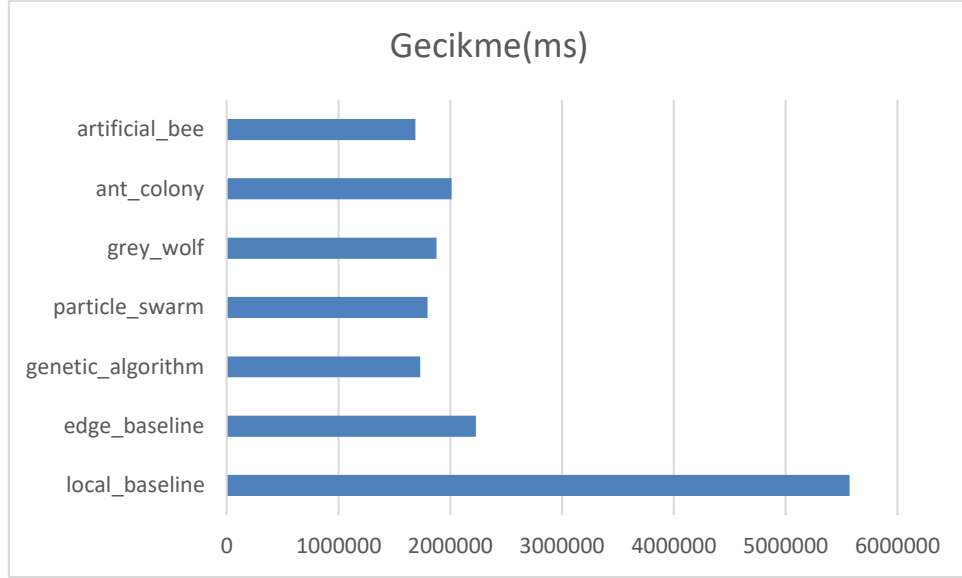
Şekil 4.15 Fiber optik veya 5G altyapısına sahip kırsal bölge için gecikme



Şekil 4.16 Fiber optik veya 5G altyapısına sahip kırsal bölge için kaynak kullanımı

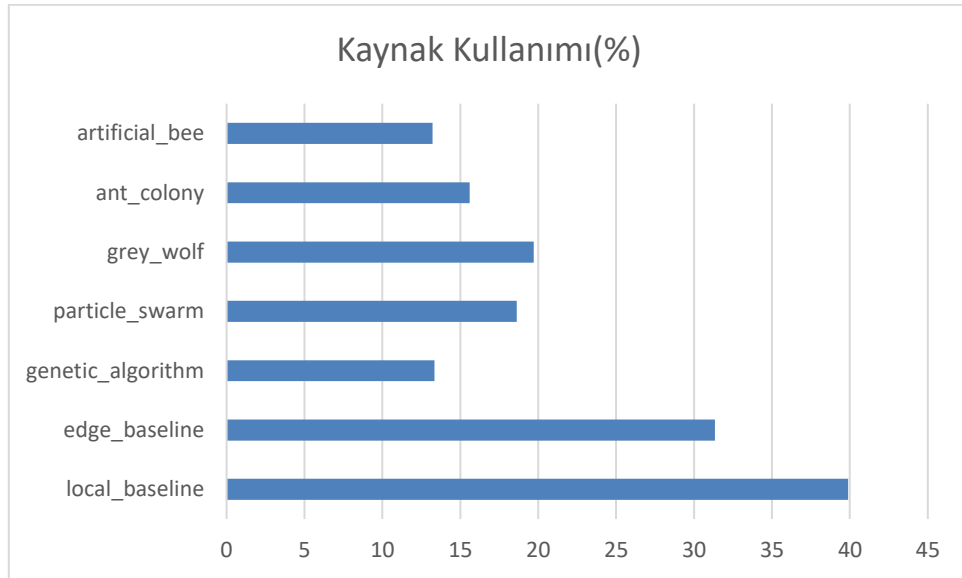
Banliyö Ofis ve Konut Ağı (Şekil 4.17–4.18): Orta yoğunluk ve dengeli kaynak profilinde

tüm algoritmalar birbirine yakın gecikme üretmiştir; belirgin bir lider çıkmamıştır (AS1). Kaynak kullanımında GA en düşük değerleri vermiştir. Referans çizgilerine göre iyileşme sınırlı fakat tutarlıdır. Dengeli profillerde algoritma seçimi, operatör farkından çok ince ayar ve protokol parametrelerine bağlı hale gelmektedir (Deb 2002).



Şekil 4.17

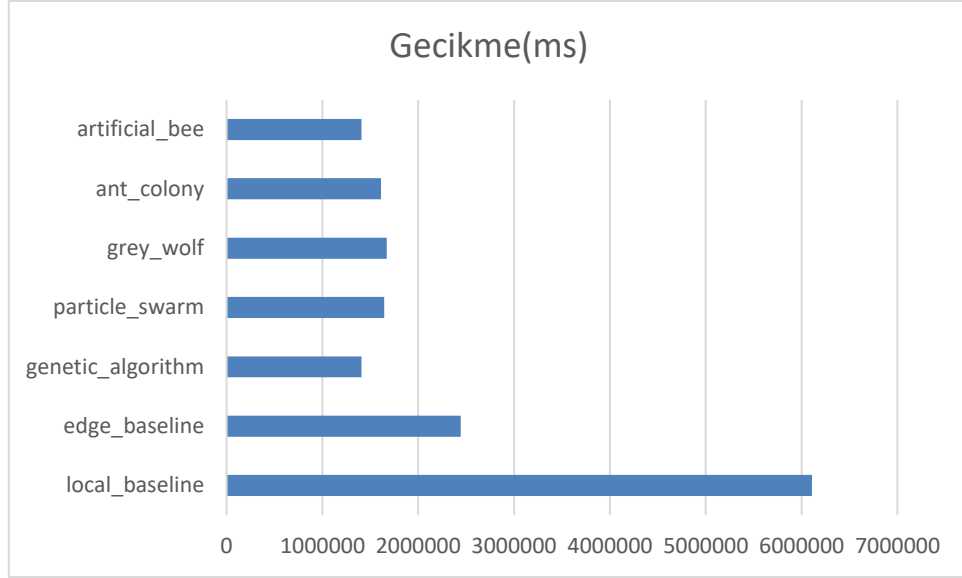
Şekil 4.17 Banliyö ofis ve konut ağı için gecikme



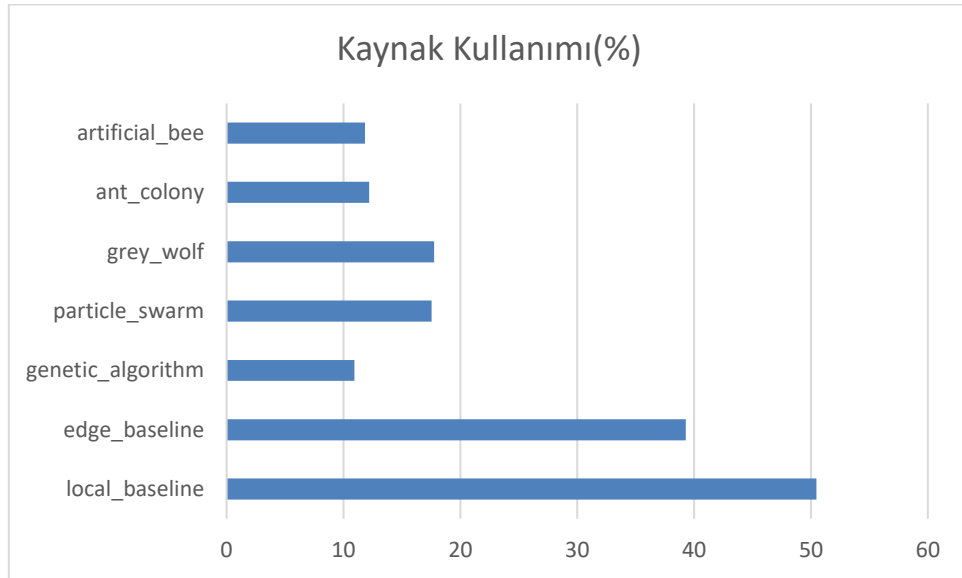
Şekil 4.18 Banliyö ofis ve konut ağı için kaynak kullanımı

Evden Çalışma ve Aile Ağı (Şekil 4.19–4.20): Orta kaynak ve karışık tüketim profilinde tüm iyileştiriciler yakın sonuç vermiştir. GA ve ABC küçük bir farkla öndedir; PSO ve

GWO kaynak kullanımında daha yüksek tüketim göstermiştir. Referans çizgilerine göre tüm algoritmalar kabul edilebilir iyileşme sağlamıştır. Ev içi ve hibrit çalışma yüklerinde limit tabanlı arama, agresif keşfe göre daha dengeli kaynak-gecikme dengesi kurmaktadır (Karaboga vd. 2007).



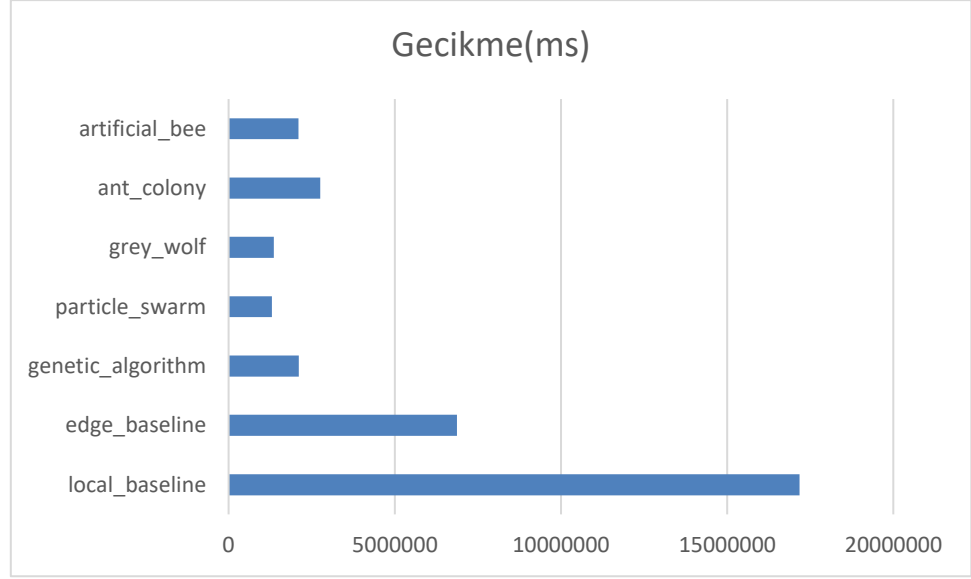
Şekil 4.19 Evden çalışma ve aile ağı için gecikme



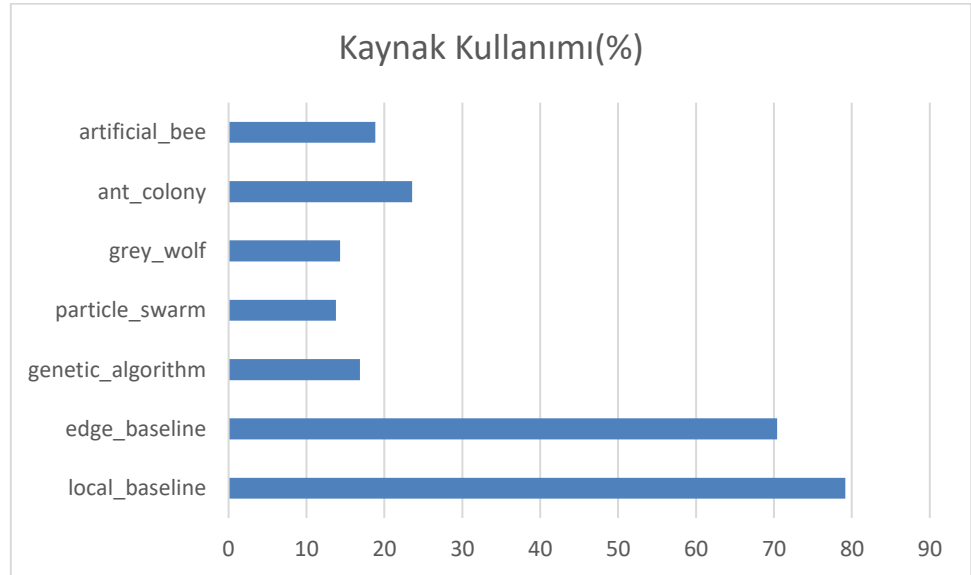
Şekil 4.20 Evden çalışma ve aile ağı için kaynak kullanımı

Büyükşehirde Veri Merkezi Performansı (Şekil 4.21–4.22): Yüksek işlem gücü ve yoğun edge kapasitesi profilinde PSO ve GWO en düşük gecikme değerini üretmiştir. GA ve

ABC birbirine yakın, ACO'ya kıyasla daha düşük gecikme gözlenmiştir. Kaynak eğrileri gecikme sıralamasıyla paralellik göstermiştir. Referans çizgilerine göre PSO/GWO bandı daha düşük bir gecikme bandında konumlanmıştır. Yüksek kaynak profillerinde agresif keşif, edge aktarımının getirisini artırmaya yöneliktir (Kennedy vd. 1995; Mirjalili vd. 2014).



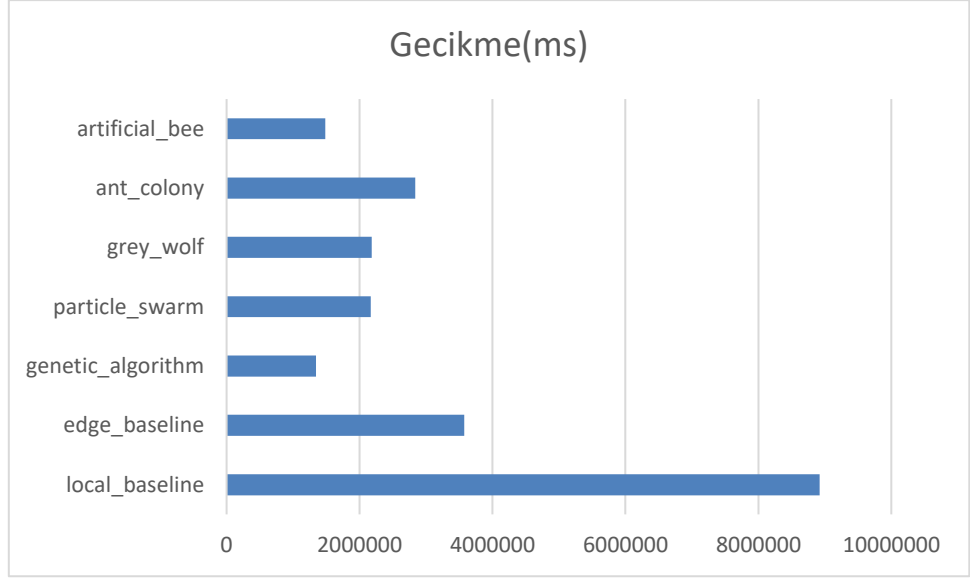
Şekil 4.21 Büyükşehirde veri merkezi performansı için gecikme



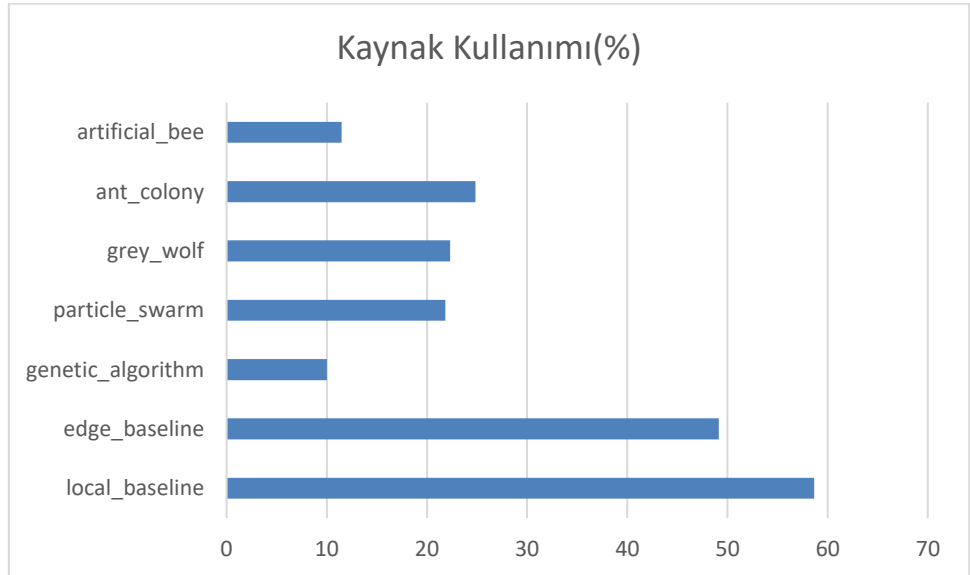
Şekil 4.22 Büyükşehirde veri merkezi performansı için kaynak kullanımı

Felaket Kurtarma Testi (Şekil 4.23–4.24): Dinamik ortam değişikliği profilinde GA en

düşük gecikme uyumunu göstermiştir; ACO en yüksek gecikme değerini üretmiştir, diğer algoritmalar bu ikisi arasında sıralanmıştır. Kaynak kullanımı gecikme ile uyumludur. Referans çizgilerine göre GA'nın görece düz eğrisi, popülasyon çeşitliliğinin dinamik koşullara adaptasyonunu desteklediğini düşündürmektedir (Goldberg 1989). Bu bulgu, dinamik senaryo grubundaki (Şekil 4.33–4.34) GA eğilimiyle örtüşmektedir.



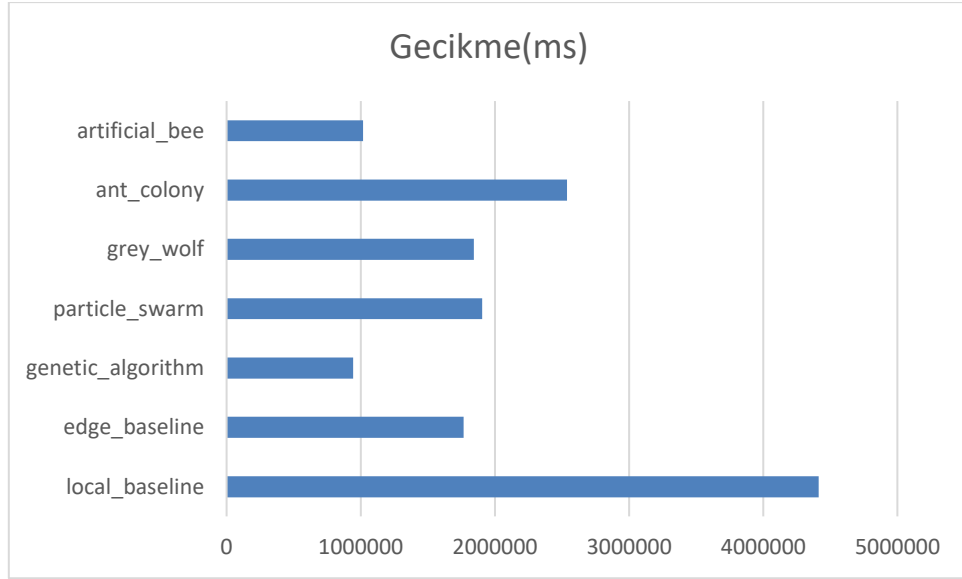
Şekil 4.23 Felaket kurtarma testi için gecikme



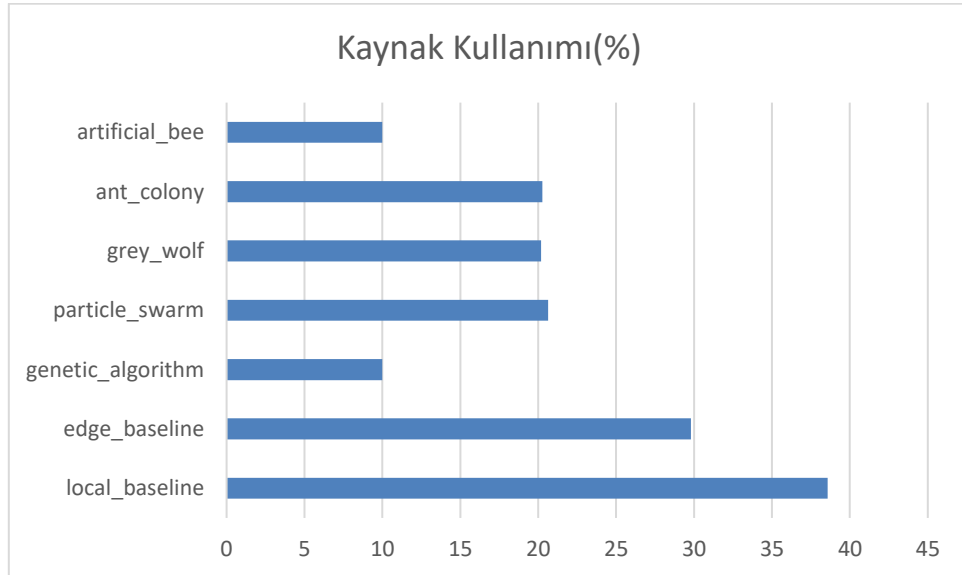
Şekil 4.24 Felaket kurtarma testi için kaynak kullanımı

Uzak Bölgelerde Uç Bilişim Ağı (Şekil 4.25–4.26): Orta-düşük kaynak ve uzak edge

profilinde GA hesaplama yükünü en dengeli biçimde dağıtmış, ABC izlemektedir. PSO ve GWO yakın performans göstermiş, ACO daha yüksek gecikme değerleri üretmiştir. Kaynak kullanımında GA ve ABC daha düşük kaynak kullanım değerleri gözlenmiştir. Uzak edge senaryolarında elitizm ve limit tabanlı arama, aşırı uç aktarımını sınırlandırarak dengeli yerleşim üretmiştir (Guo vd. 2019).



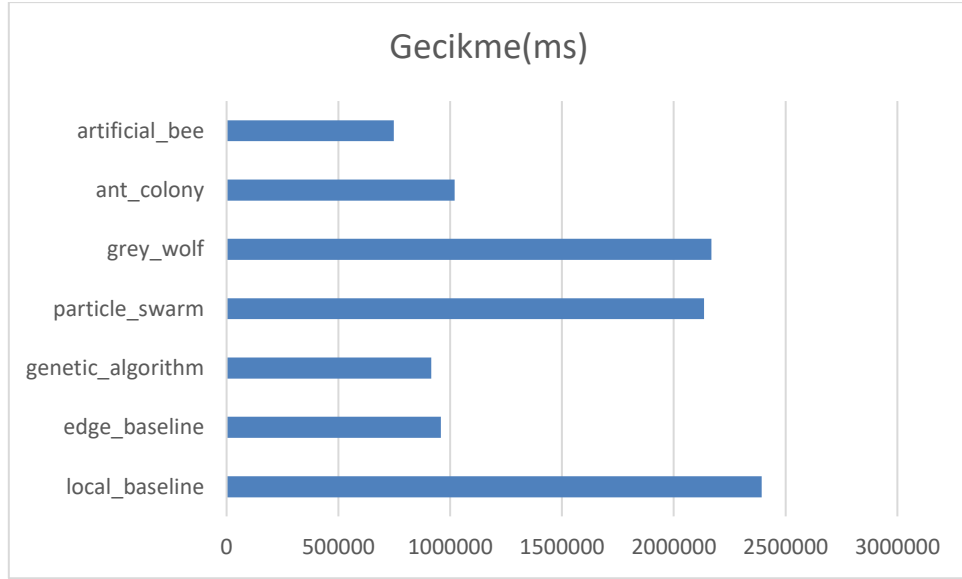
Şekil 4.25 Uzak bölgelerde uç bilişim ağı için gecikme



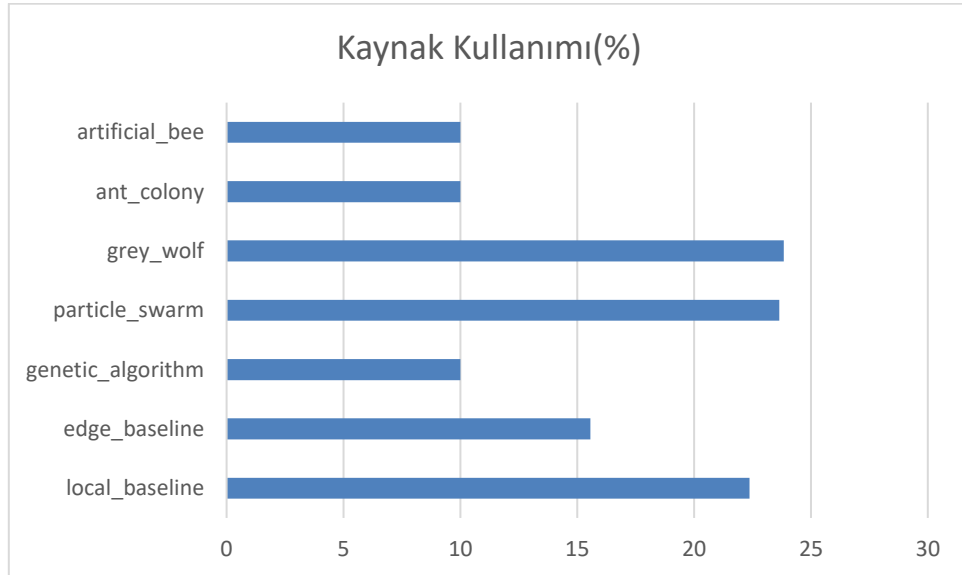
Şekil 4.26 Uzak bölgelerde uç bilişim ağı için kaynak kullanımı

Kırsal veya Uzak Alanlarda Geçici Ağ Kurulumları (Şekil 4.27–4.28): Orta-düşük

parametre profilinde GWO ve PSO daha yüksek gecikme değerleri üretmiştir. GA, ABC ve ACO birbirine yakın ve daha düşük gecikme değerleri üretmiştir. Referans çizgilerine göre agresif keşif, geçici ve kısıtlı altyapıda riskli arama uzayı genişlemesi nedeniyle daha yüksek gecikme değerleri üretmiştir. Düşük kaynak geçici kurulumlarda GA/ABC/ACO önceliği AS2 ile uyumludur.



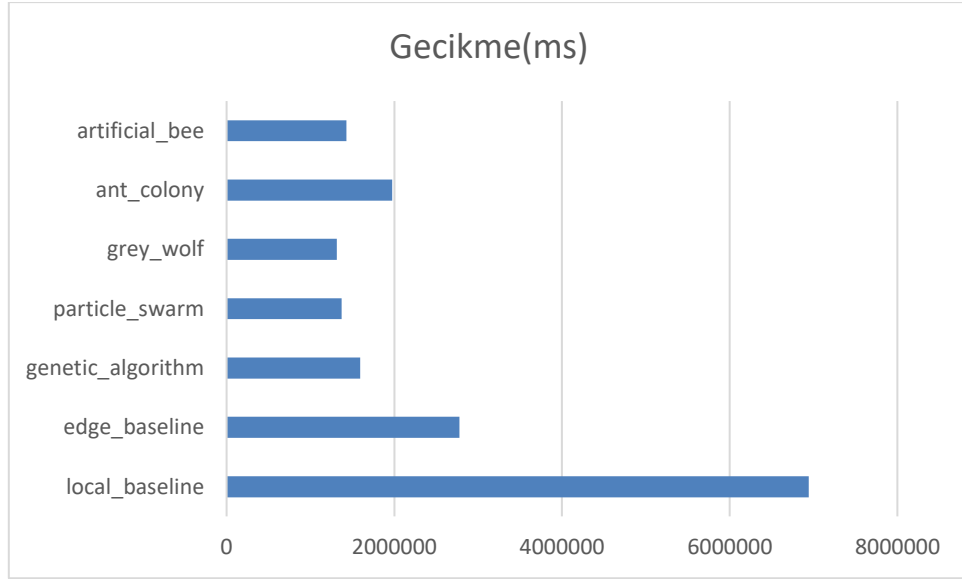
Şekil 4.27 Kırsal veya uzak alanlarda geçici ağ kurulumları için gecikme



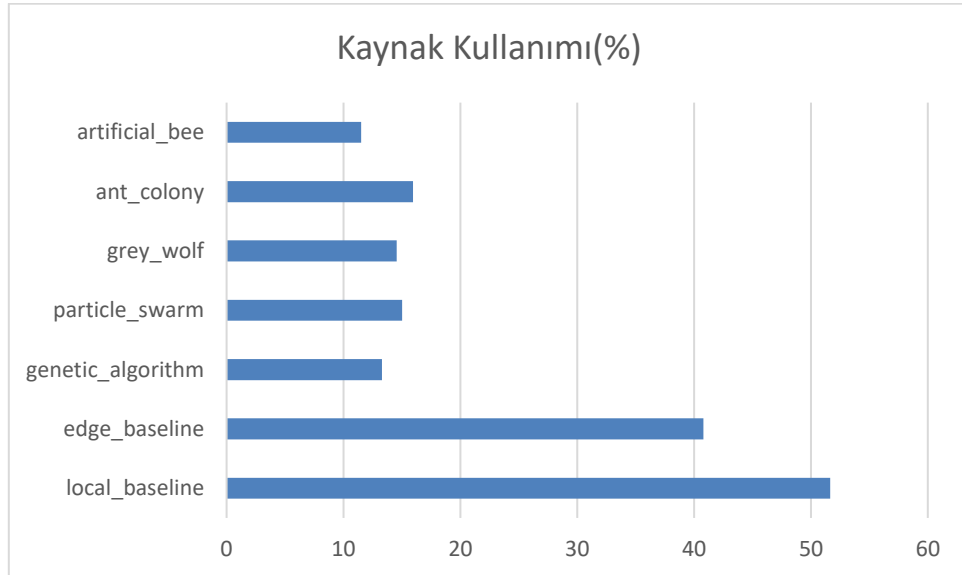
Şekil 4.28 Kırsal veya uzak alanlarda geçici ağ kurulumları için kaynak kullanımı

Kırsal Alanlarda Teletıp ve E-Öğrenme (Şekil 4.29–4.30): Yüksek ağ hızı ve yeterli

kaynak profilinde tüm iyileştiriciler birbirine yakın gecikme üretmiştir; belirgin lider çıkmamıştır. Kaynak eğrileri de benzer yakınsama göstermiştir. Geniş bant, edge aktarım cezasını düşürdüğünden operatör farkları azalmaktadır. Bu profil, Fiber/5G kırsal senaryosu ve 5G saha verisi bulgularıyla (Bölüm 4.1.3) tutarlı bir "yüksek bant – düzleşmiş sıralama" örüntüsü sunmaktadır (Liu vd. 2023).



Şekil 4.29 Kırsal alanlarda teletıp ve e-öğrenme için gecikme



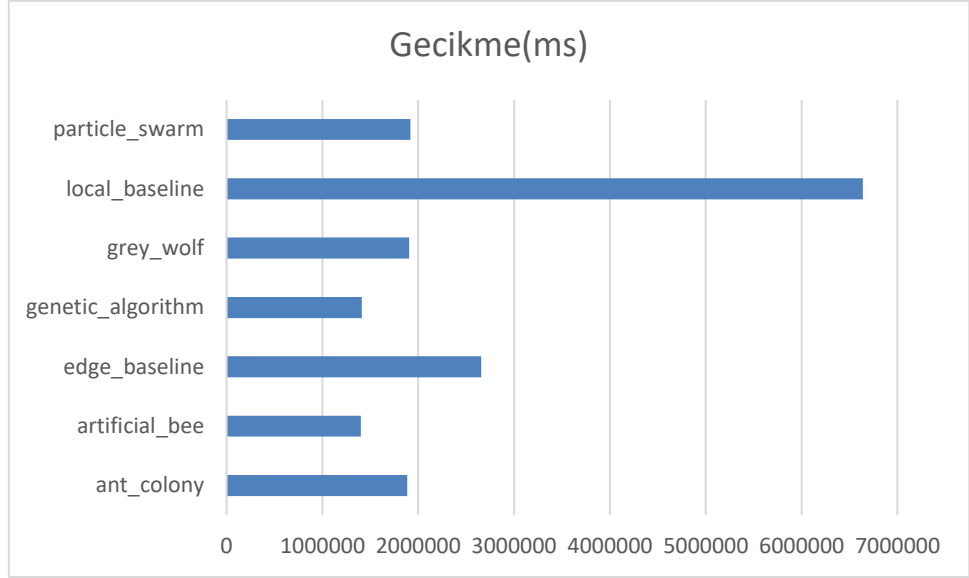
Şekil 4.30 Kırsal alanlarda teletıp ve e-öğrenme için kaynak kullanımı

Ortalama sonuçlar (Şekil 4.31–4.32): On beş senaryonun eşit ağırlıklı ortalaması

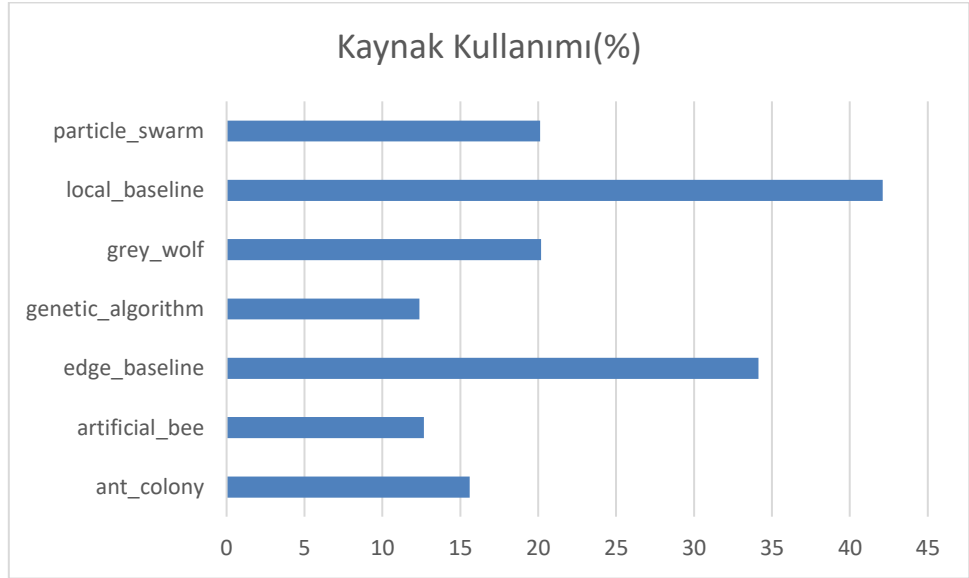
alındığında GA ve ABC, gecikme ve kaynak kullanımında diğer algoritmalara göre bir adım daha düşük gecikme değerleri üretmiştir. Bu ortalama, tüm profil sınıflarını temsil ettiğinden tekil senaryo liderliğiyle çelişebilir (AS1): örneğin yüksek yoğunluk profillerinde PSO/GWO öne çıkarken, ortalama GA/ABC lehine kayabilmektedir. Ortalama gecikme ve kaynak kullanımı Şekil 4.31–4.32'de sunulmuştur.

Dinamik ve statik senaryo grupları (Şekil 4.33–4.36) birlikte okunması gerekmektedir. Dinamik grupta (felaket kurtarma, etkinlik trafiği vb.) tüm gecikme eğrileri yükselmektedir; GA'nın görece düz kalması popülasyon çeşitliliği ile uyumludur (Goldberg 1989). Statik grupta (baz, banliyö, teletıp vb.) sıralama daha stabil kalmaktadır; GA/ABC düşük kaynak statik profillerde görece avantaj göstermektedir. PSO/GWO ise yüksek kaynak ve yoğunluk gruplarında (yüksek yoğunluklu kentsel, büyükşehir veri merkezi) referans çizgilerine göre daha düşük bir gecikme bandında konumlanmaktadır (Mirjalili vd. 2014; Kennedy vd. 1995).

Ortalama ile grup özeti arasındaki fark, algoritma seçiminde "genel ortalama" yerine "profil sınıfı" tanımının öncelikli olması gerektiğini göstermektedir (AS4). Bu profil sınıfı–algoritma eşlemesi ilgili bölümlerde tartışılmaktadır; tekil senaryo paragrafları ise istisnaları açıklamaktadır.



Şekil 4.31 Ortalama gecikme



Şekil 4.32 Ortalama kaynak kullanımı

#### 4.1.3 Gerçek dünya veri setleri ile deney sonuçları

Bu alt bölümde, beş veri setinden türetilen görev havuzları üzerinde elde edilen gerçek dünya deney sonuçları sunulmaktadır. Poisson tabanlı on beş senaryo (Bölüm 4.1) kontrollü profil taraması sağlarken, bu eksen gerçek iş yükü dağılımlarının optimizasyon davranışını test etmektedir (AS3). Önce her veri seti bağımsız (dataset-only) modda değerlendirilmekte; ardından hibrit ve adaptör+birleşik modlarla tamamlayıcı kıyaslar

raporlanmaktadır. Üç mod sırası şöyledir: (i) dataset-only: veri seti başına özgün ölçek; (ii) hibrit: sentetik ve gerçek karışım; (iii) adaptör+kalibrasyon: ortak ölçek. Tüm koşullarda Bölüm 3.5.3'te tanımlanan ortak protokol geçerlidir: aynı amaç fonksiyonu  $F(x)$ , popülasyon 100, iterasyon 100 ve yerel/uç çift yolu değerlendirmesi (Mao vd. 2017).

Birincil raporlama ölçütü uçtan uca yanıt süresi  $t_r$  (ms), kazanan yerleşim (yerel veya uç) ile simülatör gecikme ( $t_d$ ) ve yürütme ( $t_e$ ) bileşenleridir (Denklem 4.1). Tek koşuluk özetler tekli koşullardan alınmıştır;  $n=10$  tekrarlı ortalama $\pm$ std değerleri Bölüm 4.1.9 altında ayrıca verilmektedir. İkinci eksen, AS1–AS4 sorularına Poisson eksenini tamamlayıcı bulgu sunmaktadır: profil değişince lider değişmekte; entegrasyon politikası sıralamayı etkilemekte, profil tanımlandığında seçim mümkün olmaktadır (Deb 2002).

#### 4.1.4 Gerçek veri seti sonuçlarını okuma ilkeleri

Bu alt bölümdeki deneyler, Bölüm 4.1'de sentetik olarak tanımlanan on beş Poisson senaryosunu tamamlayan ikinci deneysel eksendir. Amaç, aynı karşılaştırma altyapısı altında gerçek verilerden türetilen görev gruplarının davranışını incelemektir. Beş meta-sezgisel algoritma (GA, PSO, GWO, ACO, ABC), Bölüm 3.4'te tanımlanan ortak amaç fonksiyonu  $F(x)$  ve Poisson deneyleriyle aynı popülasyon–iterasyon limitleri kullanılarak çalıştırılmıştır. Görev havuzları beş kaynaktan elde edilmiştir: Alibaba küme iş yükü (Guo vd. 2019), Azure VM yerleştirme (Hadary vd. 2020), GeoLife GPS mobilite (Zheng vd. 2010), INFOCOM'23 5G saha ölçümü (Liu vd. 2023) ve MLPerf/CNN-DailyMail çıkarım yükü (Hermann vd. 2015; Reddi vd. 2020). Kaynakların özellikleri Bölüm 2.6.2 ve 3.5.3'te açıklanmıştır.

Sonuçlar dört ana başlık altında sunulmaktadır. Birincisi, uçtan uca yanıt süresi  $t_r$  (ms). Görevin kullanıcıya dönüş süresinin birincil ölçütüdür. İkincisi, seçilen yerleşim yolu (LOCAL veya EDGE). Üçüncüsü, bu sürenin simülatör bileşenleri olan gecikme  $t_d$  (ms) ile yürütme  $t_e$  (ms). Dördüncüsü, veri setinin doğal alanlarını yansıtan ikincil görev öznitelikleri (örneğin bağlantı hızı, görev boyutu) raporlanmaktadır. Arama sırasında  $F(x)$ , kaynak tüketimi  $cr$  (%) skorunu da dikkate alır (Mao vd. 2017; Deb 2001). Buna

karşılık gerçek dünya veri setlerinde uç kapasite etiketi bulunmadığından cr değeri, ham veri ile simülatör iç yüzdesinin karıştırılmaması için sonuç özetlerinde yer almamaktadır. Poisson senaryolarında (Bölüm 4.1) hem gecikme hem kaynak birlikte raporlanır. Çünkü bu senaryolarda tüm metrikler simülatör formüllerinden ve profil tanımından doğrudan türetilmektedir (Mach vd. 2017).

Bu bölümde geçen süre kısaltmaları simülatör çıktı alanları ve Bölüm 3.4'deki gecikme ayrıştırmasıyla uyumludur. tr toplam yanıt süresidir (total\_delay ile özdeş). td iletim gecikmesi delay(x), te yürütme süresi execution(x) ile eşleştirilmiştir. Pratikte Denklem (4.1) biçiminde okunmaktadır. cr kaynak tüketimini (%) ifade etmektedir. Arama sırasında F(x)'in f<sub>2</sub> bileşeninde kullanılmaktadır; ancak gerçek kayıt tabanlı veri setlerinde uç kapasite etiketi bulunmadığından gerçek veri setlerinde kullanılmamaktadır. Kısaltmalar, tablolarda tekrarlı uzun terimler yerine tutarlı okuma sağlamak amacıyla seçilmiştir. f<sub>1</sub> ve f<sub>2</sub> amaç fonksiyonu terimleridir. tr, td ve te ise deney sonunda raporlanan süre ölçütleridir.

$$t_r \approx t_d + t_e \quad (4.1)$$

Bu bölümdeki sayılar okunurken dört husus göz önünde tutulmalıdır. (1) Mutlak milisaniye değerleri simülatör ölçeğindedir. Veri setleri arasında doğrudan toplam yapılamaz. Karşılaştırmada görelî sıralama ve yerel/uç yerleşim örüntüsü esas alınmalıdır. (2) Tek bir veri setinde tüm algoritmalar için geçerli evrensel bir kazanan yoktur (AS1). Hangi profilde hangi algoritmanın öne çıktığı ilgili sonuç tabloları ve senaryo yorumlarıyla değerlendirilmelidir. (3) Hibrit ve adaptör+birleşik modlar, farklı veri birleştirme stratejilerinin sonuçlara etkisini gösteren tamamlayıcı analizlerdir. Tezin ana bulguları yalnızca dataset-only (veri seti başına bağımsız) protokole dayanmamaktadır (Bölüm 4.1.4). (4) Azure profilinde daha önceki koşullarda bazı algoritmaların aşırı büyük gecikme ürettiği gözlenmiştir. Dataset-only koşullarında her veri setinden 150 görev kaydı kullanılmıştır. Birleşik modlarda beş kaynaktan oluşturulan 375 görevlik ortak havuz değerlendirilmiştir. Tüm deneylerde yalnızca gerçek kayıtlar kullanılmış, eksik veya uyumsuz satırlar analiz dışı bırakılmıştır (Bölüm 3.5.3).

Ham veri setlerinin tamamı deney havuzuna alınmamıştır. Önce ön işlenmiş standart tablolar kullanılmış; ardından her veri setinden rastgele alınan sabit sayıda kayıt örnek vektörüne (x) dönüştürülmüştür. Dataset-only modda bu sayı veri seti başına 150; hibrit ve adaptör+birleşik modlarda kaynak başına 75 (toplam 375 görev). Tüm koşullarda yalnızca gerçek veri setleri kullanılmış, sentetik veri devreye alınmamıştır. Bu alt örneklem hesaplama maliyeti ve tekrarlanabilirlik için bilinçli olarak dar tutulmuştur; dolayısıyla sonuçlar tüm kayıt popülasyonu üzerinden genellenebilir temsili istatistik iddiası taşımamaktadır.

#### 4.1.5 Deneysel protokolün tekrarlanabilir tanımı

Gerçek dünya deneyleri Bölüm 3.5.3–4.1.5'te tanımlanan harici veri hattı üzerinden yürütülmüştür. Her koşuda veri seti yüklenmiş, eksik kayıtlar elenmiş, örnek vektörleri adaptör ile üretilmiş ve simülatör sınırlarına hizalanmıştır. Her aday çözüm için yerel ve uç yürütme yolları ayrı ayrı değerlendirilmiş; beş meta-sezgisel algoritma aynı görev havuzu üzerinde karşılaştırılmıştır. Ayrıntılı protokol Çizelge 4.1'de özetlenmektedir.

Çizelge 4.1 Harici deney protokolü

| Parametre                 | Dataset-only | Hibrit    | Adaptör+birleşik |
|---------------------------|--------------|-----------|------------------|
| Satır/kayıt veya havuz    | 150/kayıt    | 375 havuz | 375 havuz        |
| Require_real_data         | Evet         | Evet      | Evet             |
| Use_minmax_calibration    | Hayır        | Evet      | Evet (global)    |
| Clamp_to_simulator_bounds | Evet         | Evet      | Evet             |
| A (hibrit)                | —            | 0,62      | —                |
| Blend_seed                | —            | 7         | —                |
| Popülasyon / iterasyon    | 100 / 100    | 100 / 100 | 100 / 100        |

Simülatör çıktıları milisaniye (ms) cinsinden raporlanmaktadır (Mao vd. 2017; Shi vd. 2016; Liu vd. 2020). Dataset-only yanıt süresi sonuçları Çizelge 4.2'de raporlanmıştır.

Dataset-Only Modu: Yanıt Süresi Sonuçları (Çizelge 4.2–4.3)

Çizelge 4.2 Dataset-only: en iyi bireyin yanıt süresi tr (ms)

| Kayıt (etiket)    | GA     | PSO   | GWO   | ACO     | ABC     | En düşük tr  |
|-------------------|--------|-------|-------|---------|---------|--------------|
| MLPERF_EDGE       | 223,19 | 12,82 | 12,65 | 1163,08 | 1163,08 | GWO          |
| 5G_MONARCH        | 14,88  | 11,77 | 12,28 | 373,55  | 18,96   | PSO          |
| CRAWDAD (GeoLife) | 16,62  | 12,82 | 12,65 | 19,34   | 19,07   | GWO          |
| ALIBABA_V2        | 22,85  | 12,28 | 12,28 | 40,67   | 12,28   | GWO/PSO /ABC |
| AZURE_PUBLIC      | 12,82  | 12,28 | 17,25 | 13,24   | 13,24   | PSO          |

Not: Çizelge 4.2 dataset-only tek koşu değerleridir; n=10 seed tekrarlı ortalama±std için Bölüm 4.1.9 ve Çizelge 4.7'ye bakınız. Kazanan yerleşim dağılımı Çizelge 4.3'te verilmiştir.

Çizelge 4.3 Dataset-only: kazanan yerleşim (LOCAL / EDGE)

| Kayıt (etiket) | GA    | PSO   | GWO   | ACO   | ABC   |
|----------------|-------|-------|-------|-------|-------|
| MLPERF_EDGE    | LOCAL | LOCAL | LOCAL | LOCAL | LOCAL |
| 5G_MONARCH     | LOCAL | LOCAL | EDGE  | LOCAL | LOCAL |
| CRAWDAD        | LOCAL | LOCAL | LOCAL | LOCAL | EDGE  |
| ALIBABA_V2     | EDGE  | EDGE  | EDGE  | EDGE  | EDGE  |
| AZURE_PUBLIC   | LOCAL | EDGE  | LOCAL | LOCAL | LOCAL |

Çizelge 4.4 Dataset-only: gecikme ve yürütme bileşenleri (ms)

| Veri Seti       | Bleşn | GA     | PSO   | GWO   | ACO     | ABC     |
|-----------------|-------|--------|-------|-------|---------|---------|
| MLPERF<br>EDGE  | td    | 1,22   | 1,27  | 1,10  | 1,27    | 1,27    |
| MLPERF<br>EDGE  | te    | 221,96 | 11,55 | 11,55 | 1161,81 | 1161,81 |
| 5G<br>MONARCH   | td    | 1,14   | 1,16  | 1,00  | 1,50    | 1,16    |
| 5G<br>MONARCH   | te    | 13,74  | 10,61 | 11,28 | 372,06  | 17,80   |
| CRAWDAD         | td    | 1,22   | 1,27  | 1,10  | 1,27    | 1,00    |
| CRAWDAD         | te    | 15,40  | 11,55 | 11,55 | 18,07   | 18,07   |
| ALIBABA V2      | td    | 1,00   | 1,00  | 1,00  | 1,00    | 1,00    |
| ALIBABA V2      | te    | 21,85  | 11,28 | 11,28 | 39,67   | 11,28   |
| AZURE<br>PUBLIC | td    | 1,01   | 1,00  | 1,50  | 1,01    | 1,01    |
| AZURE<br>PUBLIC | te    | 11,81  | 11,28 | 15,75 | 12,23   | 12,23   |

Çizelge 4.4, Çizelge 4.2'deki toplam yanıt süresinin td ve te bileşenlerine ayrılmış halini sunmaktadır. Gecikme ve yürütme bileşenleri birlikte okunduğunda şu patern görülmektedir: iletim gecikmesi (td) beş veri setinde de yaklaşık 1–1,5 ms aralığında çıkmakta ve algoritmalar arasında belirgin bir biçimde ayrılmamaktadır. Buna karşılık toplam yanıt süresindeki farkların büyük bölümü yürütme süresi (te) üzerinden oluşmaktadır. Başka bir deyişle, hangi algoritmanın daha iyi performans gösterdiği esas olarak görevin yerelde mi uçta mı yürütüleceğine bağlıdır. İletim tarafı bu koşul setinde ikincil kalmaktadır (Mao vd. 2017; You vd. 2017). Bileşen değerleri Çizelge 4.4'te verilmiştir.

#### 4.1.6 Veri seti başına ayrıntılı bulgu ve literatür yorumu

MLPerf / CNN-DailyMail (MLPERF\_EDGE): GWO (12,65 ms) ve PSO (12,82 ms), GA'dan (223,19 ms) ve ACO/ABC'den (1163,08 ms) belirgin şekilde daha iyi sonuç vermiştir. Kazanan çözümlerde GWO/PSO görev büyüklüğü $\approx 10$  ve karmaşıklık $\approx 1$ 'e yakınsamaktadır; GA task\_size $\approx 67,86$  ile yüksek te üretmektedir; ACO/ABC task\_size $\approx 143,96$  ile ağır kuyruk göstermektedir (Çizelge 4.4). Tüm algoritmalar yerel yürütmeyi seçmiştir. Bu veri setindeki davranış yüksek yoğunluklu kentsel senaryosu

benzerlik göstermektedir (Şekil 4.3–4.4). Sürü keşfi kompakt görev yapılandırmasına yakınsamakta; Feromon araması geniş görev boyutunda kötü sonuç vermektedir. (Kennedy vd. 1995; Mirjalili vd. 2014; Dorigo vd. 2006; Reddi vd. 2020).

INFOCOM'23 5G (5G\_MONARCH: En düşük tr PSO'da (11,77 ms, yerel) elde edilmiştir. GWO uç'ta 12,28 ms ile birim gecikme avantajı kullanılmaktadır. ACO 373,55 ms ile yavaş kalmıştır ( $t_{\approx 372}$  ms). Liu vd. 2023, bağlantı kalitesi değişkenliğini vurgulamaktadır; dual-path değerlendirme kanal koşullarına göre yerel/uç seçimine izin vermektedir (Mao vd. 2017). GA (14,88 ms) orta düzeyde rekabetçidir; ABC (18,96 ms) yerel kalmaktadır. Bu veri setindeki davranış Fiber/5G kırsal senaryo ile benzerdir (Şekil 4.15–4.16). Çoklu testte PSO liderliği karardır (Çizelge 4.7).

GeoLife mobilite (CRAWDAD), GPS yörünge profili: GWO ve PSO  $\sim 12,65$ – $12,82$  ms bandındadır; ACO/ABC  $\sim 19$  ms. ABC tek uç seçmiş, diğerleri yerel yürütmeyi tercih etmiştir. Görev boyutu veride zayıf olduğundan adaptör projeksiyonu kritik rol oynamaktadır (Bölüm 3.5.4). Mobilite ağırlıklı veri setlerde sürü optimizasyonları düşük gecikme bandına yakınsamaktadır (Zheng vd. 2010). Bu veri setindeki davranış akıllı şehir/orta yoğunluk profilli senaryolarla benzerdir. Çoklu testte GWO en düşük ortalama değeri vermiştir (Çizelge 4.7).

Alibaba v2018 (ALIBABA\_V2), veri merkezi kaynak profili: GWO, PSO ve ABC 12,28 ms'te eşittir; GA 22,85 ms; ACO 40,67 ms. Tüm algoritmalar uç yürütmeyi seçmiştir; yüksek edge kaynak kullanılabilirliği uç aktarımını desteklemektedir (Guo vd. 2019). GA'nın yüksek görev boyutu (19,37) ile yavaşlaması elitizmin kompakt çözüme yakınsamamasından kaynaklanabilir (Goldberg 1989). Bu veri setindeki davranış büyükşehir veri merkezi performansına benzemektedir (Şekil 4.21–4.22).

Azure VM packing (AZURE\_PUBLIC), VM yerleşim profili: Simülatör sınır kırpması etkinleştirildiğinde (clamp\_to\_simulator\_bounds=True; Bölüm 3.5.6) tüm algoritmalar 12–17 ms bandına sıkışmaktadır; PSO en düşük (12,28 ms, uç). Kırpma kapalıyken erken tekli koşuda GA ölçek patlaması gözlenmiş; güncel protokolle engellenmiştir (Hadary vd. 2020). Dar bant, güvenli arama uzayına hizalanmanın önemini doğrulamaktadır (Ben-

David vd. 2010). Çoklu testte yüksek std, seed duyarlılığını göstermektedir (Çizelge 4.7). Bu veri setindeki davranış banliyö/dengeli profiline uygundur (Şekil 4.17–4.18).

#### 4.1.7 Birleşik modlar: hibrit ve adaptör+kalibrasyon

Birleşik mod sonuçları Çizelge 4.5'te özetlenmiştir. İki mod, AS3 kapsamında entegrasyon politikasının algoritma sıralamasını nasıl değiştirdiğini test etmektedir: hibrit karışım (Sohaib vd. 2024) ve adaptör+global kalibrasyon (Ben-David vd. 2010). Tekli koşu özetleri yön göstermekte; çoklu test özetleri (Çizelge 4.9) seed duyarlılığını ortaya çıkarmaktadır.

Çizelge 4.5 Birleşik modlar: tr (ms) ve yerleşim

| Mod                     | GA     | PSO   | GWO   | ACO         | ABC    | En düşük | Yerleşim (en iyi) |
|-------------------------|--------|-------|-------|-------------|--------|----------|-------------------|
| Hibrit<br>$\alpha=0,62$ | 151,45 | 11,97 | 11,66 | 1220,1<br>7 | 259,24 | GWO      | LOCAL             |
| Adaptör+<br>birleşik    | 13,93  | 11,64 | 13,93 | 18,93       | 11,62  | ABC      | ABC:<br>LOCAL     |

Hibrit modda beş veri kaynağından türetilen kayıtlar sentetik görev üretimiyle karıştırılarak ortak havuz oluşturulmuştur (Sohaib vd. 2024). Tek koşuda GWO (11,66 ms) ve PSO (11,97 ms) önde; ACO 1220,17 ms ile belirgin yüksek kalmıştır. Çoklu testte GWO  $12,80 \pm 0,74$  ms ile lider çıkmıştır. PSO  $53,9 \pm 124,7$  ms ile yüksek std göstermiştir. ACO'nun MLPerf benzeri geniş görev boyutu bölgelerinde takılması, hibrit havuzdaki ağır kuyruk kayıtlarını açıklamaktadır (Çizelge 4.9; Dorigo vd. 2006).

Adaptör + global kalibrasyon modunda beş kaynak ortak ölçeğe hizalanarak birleştirilmiştir (Ben-David vd. 2010). Tek koşuda ABC (11,62 ms) önde; tekrarlı koşuda GWO ( $13,01 \pm 1,20$  ms) ve PSO ( $13,51 \pm 0,88$  ms) yakın performans göstermiştir. GA dar bantta ( $13,94 \pm 0,04$  ms) kalmış; ACO yüksek ortalama ve geniş sapma üretmiştir ( $55,7 \pm 90,5$  ms). Kalibrasyonlu havuz sıralamayı düzeltmektedir: dataset-only moda göre algoritma farkları azalmaktadır. Bu, ortak ölçek politikasının profil ayrımını kısmen gizleyebileceğini göstermektedir (AS3).

#### 4.1.8 Tekrarlı koşu protokolü ve sonuçları

Meta-sezgisel arama stokastik bir süreçtir: aynı görev havuzu ve aynı arama limitleri kullanılsa bile en iyi çözüm, başlangıç popülasyonu değişikçe farklılaşabilmektedir (Goldberg 1989; Kennedy vd. 1995; Mirjalili vd. 2014). Bu nedenle tek koşuluk sonuçlar yön gösterici kanıt niteliğindedir;  $n \geq 10$  bağımsız tekrar ile ortalama $\pm$ std hesaplanmıştır (Deb 2002; Rao 2009) (Bölüm 4.1.11). Simülatör katmanı deterministiktir: aynı x ve aynı yerleşim yolu için gecikme skorları tekrarlanır (Bölüm 3.4.2; Mao vd. 2017). Belirsizlik optimizasyon döngüsündedir. Dolayısıyla tekrarlı koşularda elde edilen ortalama ve standart sapma değerleri, algoritma performansının daha güvenilir göstergesi olarak değerlendirilmektedir. Ortalama farkın küçük, standart sapmanın büyük olduğu durumlarda algoritma sıralaması kararsız kabul edilmektedir. Bu bulgu AS1 hipotezini desteklemektedir (Wilcoxon 1945) (Bölüm 4.1.5).

Gerçek dünya verileri için tekrarlı deneyler Bölüm 4.1.9'de tanımlanan protokolle yürütülmüştür. Dataset-only ve birleşik mod özetleri aşağıda sunulmakta; koşu parametreleri Çizelge 4.6'da özetlenmektedir.

Çizelge 4.6 Veri seti kökenli tekrarlı koşu parametreleri

| Parametre                        | Önerilen değer                | Gerekeçe                                    |
|----------------------------------|-------------------------------|---------------------------------------------|
| N_repeats                        | $\geq 10$<br>(tercihen 20–30) | Merkezi limit teoremi için yeterli örnek    |
| Base_seed                        | 42 (sabit, raporlanır)        | Tekrarlanabilirlik                          |
| N_rows                           | 150/kayıt                     | Tekli koşularla hizalı                      |
| Population_size / max_iterations | 100 / 100                     | Poisson ve veri seti batch ile aynı         |
| Require_real_data                | True                          | Proxy/sentetik düşme engellenir             |
| Use_minmax_calibration           | False<br>(dataset-only)       | Kayıtbaşına ölçek korunur (Guo vd. 2019)    |
| Clamp_to_simulator_bounds        | True                          | Simülatör taşması önlenir (Hadary vd. 2020) |

#### 4.1.9 İstatistiksel raporlama ve wilcoxon testi

Deneyisel sonuçların bu tezde iki katmanda okunması gerekmektedir. Birinci katman tekli koşullardır: her (veri seti, algoritma) çifti yalnızca bir kez çalıştırılmıştır ve hangi profilde hangi algoritmanın öne çıktığına dair ilk ipuçlarını vermektedir (Çizelge 4.2). İkinci katman çoklu test koşullarıdır: aynı protokol on kez, farklı başlangıç değerleriyle yürütülmüştür. Sonuçlar ortalama±std ile özetlenmiştir. Dataset-only tekrarlar Çizelge 4.7'de. Hibrit ve adaptör+birleşik tekrarları Çizelge 4.9'da yer almaktadır. Tek koşuluk birleşik mod özetleri ise Çizelge 4.5'tedir. Okuma sırası şöyledir: önce Çizelge 4.2 ve 4.5 ile yön belirlenmektedir. Ardından Çizelge 4.7 ve 4.9 ile bu yön çoklu test istatistikleriyle doğrulanmaktadır. Bu çerçevede, Poisson senaryoları (Bölüm 4.1) ile birlikte okunduğunda tamamlanmaktadır. Çoklu test özetleri (Çizelge 4.7–4.9) metinde özetlenen algoritma önerilerini desteklemekte veya sınırlandırmaktadır (Mach vd. 2017; Taleb vd. 2017).

Eşleştirilmiş on tekrar çıktıları için Wilcoxon işaretli sıra testi tasarlanmış; Çizelge 4.8, Çizelge 4.7 özetlerinden türetilen ikili karşılaştırma yorumunu sunmaktadır (Wilcoxon 1945). Çizelge 4.8'deki ikili karşılaştırmalar şu biçimde okunması önerilmektedir: (i) PSO–GWO çiftlerinde farklar çoğu profilde istatistiksel olarak anlamlı değildir; bu durum, iki sürü algoritmasının benzer keşif karakterini yansıtmaktadır. (ii) MLPerf ve 5G profillerinde ACO/ABC ile PSO/GWO arasındaki ayrışma belirgindir; büyük görev boyutu ve feromon maliyeti bu ayrımı açıklamaktadır (Dorigo vd. 2006; Reddi vd. 2020). (iii) Anlamlılık, pratik anlamlılıkla birlikte değerlendirilmesi gerekmektedir: birkaç ms fark düşük std ile desteklenmekte ise kararlı üstünlük; yüzlerce ms std ile desteklenmekte ise seed duyarlılığı olarak yorumlanması gerekmektedir. Bonferroni düzeltmesi ( $\alpha'=0,005$ ) çoklu ikili karşılaştırmalarda FWER( Family-Wise Error Rate) kontrolü sağlamaktadır (Wilcoxon 1945).

Poisson senaryo sonuçları (Bölüm 4.1) ile gerçek dünya veri seti sonuçları (Bölüm 4.1.3) rakip değil, tamamlayıcı bulgu kaynaklarıdır. Poisson ekseninde on beş farklı  $\lambda$  profiliyle kontrollü bir tarama yapılmıştır: yoğunluk ve kaynak koşulları sistematik biçimde değiştirilmiş, hangi algoritma ailesinin hangi koşulda öne çıktığı görülmüştür. Veri seti ekseninde ise beş gerçek veri seti (Alibaba, Azure, GeoLife, 5G saha, MLPerf)

kullanılmıştır. Burada iş yükü dağılımı sentetik üretici yerine veri kayıtlarından gelmektedir. İki eksen birlikte değerlendirildiğinde, simülasyonda görülen eğilimlerin gerçek kayıt tabanlı koşullarda tekrarlanıp tekrarlanmadığı ve hangi profillerde sapma olduğu daha sağlıklı yorumlanabilir (Guo vd. 2019; Hadary vd. 2020).

#### 4.1.10 Dataset-only çoklu test sonuçları

Bu alt bölümde, beş veri setinde dataset-only modda yürütülen çoklu test koşullarının özeti verilmektedir. Her (veri seti, algoritma) çifti on bağımsız tekrarla çalıştırılmıştır; toplam 250 koşu ( $5 \text{ veri seti} \times 5 \text{ algoritma} \times 10 \text{ tekrar}$ ) tamamlanmıştır. Sonuçlar Çizelge 4.7'de ortalama $\pm$ std olarak gösterilmektedir. Çoklu test koşullarında protokol, tekli dataset-only koşullarıyla aynı tutulmuştur: yalnızca gerçek veri kullanılmış, veri seti başına min–max kalibrasyon kapalı bırakılmış ve simülatör sınır kırpması etkin olmuştur (Bölüm 4.1.5). Tek fark, her tekrarda başlangıç değerinin değiştirilmesidir.

Çizelge 4.7'ye göre en düşük ortalama tr veri setinden veri setine değişmektedir: 5G ve MLPerf'te PSO, GeoLife, Alibaba ve Azure'da GWO öne çıkmaktadır. Tek bir algoritma tüm veri setlerinde lider değildir (AS1). PSO ve GWO, dört sette yaklaşık 11,77–16,50 ms bandında ve düşük standart sapmayla kararlıdır. GA orta düzeyde kalırken ACO, 5G ve Alibaba'da yüzlerce ms; MLPerf'te ise ~1335 ms ortalama üretmektedir — bu değer tekli koşudaki 1163 ms ile uyumludur (Reddi vd. 2020; Dorigo vd. 2006).

Çizelge 4.7 Dataset-only: tr ortalama $\pm$ std (ms)

| İz (etiket)        | GA                  | PSO              | GWO              | ACO                 | ABC                 | En düşük (ort.) |
|--------------------|---------------------|------------------|------------------|---------------------|---------------------|-----------------|
| MLPERF_E DGE       | 309,54 $\pm$ 106,03 | 12,82 $\pm$ 0,00 | 12,84 $\pm$ 0,31 | 1335 $\pm$ 220      | 1167 $\pm$ 3        | PSO             |
| 5G_MONA_RCH        | 14,53 $\pm$ 0,49    | 11,77 $\pm$ 0,01 | 11,88 $\pm$ 0,34 | 436 $\pm$ 246       | 18,96 $\pm$ 0,00    | PSO             |
| CRAWDA D (GeoLife) | 16,50 $\pm$ 0,40    | 12,82 $\pm$ 0,01 | 12,78 $\pm$ 0,27 | 19,34 $\pm$ 0,01    | 19,20 $\pm$ 0,14    | GWO             |
| ALIBABA_V2         | 18,14 $\pm$ 6,39    | 12,32 $\pm$ 0,09 | 12,22 $\pm$ 0,21 | 177 $\pm$ 123       | 17,96 $\pm$ 12,0    | GWO             |
| AZURE_P UBLIC      | 8143 $\pm$ 13 168   | 1638 $\pm$ 2104  | 14,19 $\pm$ 2,50 | 48 175 $\pm$ 67 804 | 24 423 $\pm$ 46 850 | GWO             |

Çizelge 4.8, dataset-only modda birinci ve ikinci grup algoritmalar arasındaki eşleştirilmiş Wilcoxon karşılaştırmalarını özetlemektedir. Testler aynı seed dizisindeki on tekrar çıktısı üzerinde yürütülmüştür; Bonferroni düzeltilmiş eşik  $\alpha'=0,005$  (m=10 ikili karşılaştırma) kullanılmıştır.

Çizelge 4.8 Dataset-only modda eşleştirilmiş Wilcoxon testi özeti (n=10)

| Veri seti | Karşılaştırma | $\Delta\mu$ (ms) | p (yakl.) | $\alpha'=0,005$ | Yorum          |
|-----------|---------------|------------------|-----------|-----------------|----------------|
| MLPerf    | PSO–GWO       | 0,02             | >0,05     | Hayır           | Anlamalı değil |
| MLPerf    | PSO–ACO       | 1322,18          | <0,001    | Evet            | PSO üstün      |
| MLPerf    | PSO–ABC       | 1154,18          | <0,001    | Evet            | PSO üstün      |
| 5G        | PSO–GWO       | 0,11             | >0,05     | Hayır           | Anlamalı değil |
| 5G        | PSO–ACO       | 424,23           | 0,003     | Evet            | PSO üstün      |
| GeoLife   | GWO–PSO       | 0,04             | >0,05     | Hayır           | Anlamalı değil |
| GeoLife   | GWO–ACO       | 6,56             | <0,001    | Evet            | GWO üstün      |
| Alibaba   | GWO–PSO       | 0,10             | 0,089     | Hayır           | GWO üstün      |
| Alibaba   | GWO–ACO       | 164,78           | 0,028     | Hayır           | GWO üstün      |
| Azure     | GWO–PSO       | 1623,81          | 0,028     | Hayır           | GWO üstün      |

Sütun p, ortalama $\pm$ std özetlerinden hesaplanan etki büyüklüğüne dayalı yaklaşık sınamadır; kesin p-değerleri ham tekrar çıktılarıyla doğrulanmalıdır. PSO/GWO gibi yakın çiftlerde fark beklenen biçimde anlamlı değildir; MLPerf ve 5G profillerinde ACO/ABC ile PSO/GWO ayrışması belirgindir.

Azure profilinde tekli koşu ile çoklu test sonuçları arasında belirgin fark vardır. Tekli koşuda (Çizelge 4.2) tüm algoritmalar 12,28–17,25 ms aralığındaydı; çoklu testte yalnızca GWO 14,19 $\pm$ 2,50 ms ile dar bantta kalmış, diğer algoritmalar binlerce ms ortalamaya çıkmıştır (GA 8143 ms, PSO 1638 ms, ACO 48 175 ms, ABC 24 423 ms). Bu bulgu, Azure profilinde başlangıç değerine bağlı sonuç değişkenliğinin yüksek olduğunu ve tekli koşunun tek başına yeterli kanıt sunmadığını göstermektedir (Hadary vd. 2020).

MLPerf profilinde GA tekli koşuda 223,19 ms iken çoklu test ortalaması 309,54 $\pm$ 106,03 ms'e yükselmiştir; PSO ve GWO ise ~12,82 ms bandında sabit kalmıştır (std $\approx$ 0). Çıkarım yükü profillerinde PSO/GWO önceliği çoklu testle güçlenirken GA, ACO ve ABC zayıf kalmaktadır (AS2). Çizelge 4.7, tekli koşu bulgularının çoklu test ortalamalarıyla nasıl

teyit edildiğini veya sınırlandığını göstermektedir. Algoritmalar arası üstünlük değerlendirmeleri Çizelge 4.8'deki eşleştirilmiş Wilcoxon testleriyle desteklenmektedir.

#### 4.1.11 Birleşik mod çoklu test sonuçları

Bu alt bölümde, beş veri kayıtlarının birleştirildiği iki modda yürütülen çoklu test koşullarının özeti verilmektedir: hibrit ve adaptör+birleşik. Her mod için on bağımsız tekrar tamamlanmıştır. Ortak havuz 375 görevden oluşmuştur. Sonuçlar Çizelge 4.9'da ortalama±std olarak raporlanmaktadır. Protokol, tekli birleşik mod koşullarıyla (Çizelge 4.5) aynı tutulmuştur; tek fark her tekrarda başlangıç kümesinin değiştirilmesidir.

Birleşik mod sonuçları, tekli koşu özetleri (Çizelge 4.5) ile birlikte okunması gerekmektedir. Hibrit modda GWO 12,80±0,74 ms ile dar bantta kalmış ve en düşük ortalamaı vermiştir. PSO'nun yüksek std'si (53,9±124,7 ms) ve GA/ABC/ACO'nun yüzlerce ms bandı, hibrit havuzda seed duyarlılığının yüksek olduğunu göstermektedir. Tekli koşuda ACO 1220 ms iken çoklu test ortalaması 267 ms'e düşmüştür; yine de GWO'dan kötü kalmıştır. Düşük std (GWO hibrit/adaptör) kararlı liderliği; yüksek std (PSO hibrit) tekli koşuya güvenilmemesi gerektiğini göstermektedir.

Çizelge 4.9 Birleşik modlar: tr ortalama±std (ms)

| Mod                         | GA                | PSO               | GWO             | ACO               | ABC               | En düşük (ort.) |
|-----------------------------|-------------------|-------------------|-----------------|-------------------|-------------------|-----------------|
| Hibrit<br>( $\alpha=0,62$ ) | 161,51 ±<br>20,64 | 53,86 ±<br>124,73 | 12,80 ±<br>0,74 | 266,72 ±<br>10,88 | 251,44 ±<br>11,42 | GWO             |
| Adaptör +<br>birleşik       | 13,94 ±<br>0,04   | 13,51 ±<br>0,88   | 13,01 ±<br>1,20 | 55,69 ±<br>90,46  | 15,25 ±<br>3,83   | GWO             |

Adaptör+birleşik modda çoklu test lideri GWO olmuştur (13,01±1,20 ms). Tekli koşuda ABC'nin öne çıkması (11,62 ms) tekrarlarda sürdürülmemiştir; bu durum, birleşik havuzda sıralamanın seed'e bağlı olabileceğini göstermektedir. Mod başına farklı lider çıkması AS1 ile, entegrasyon politikasının sıralamayı değiştirmesi AS3 ile uyumludur.

Uygulama açısından birleşik mod seçiminin hedef profilin ölçek homojenliğine göre yapılması önerilmektedir: heterojen veri setleri için adaptör+kalibrasyon, sentetik tamamlama için hibrit (Sohaib vd. 2024; Ben-David vd. 2010).

#### **4.1.12 Poisson senaryoları ile gerçek dünya veri seti sonuçlarının çapraz karşılaştırması**

Poisson senaryo sonuçları (Bölüm 4.1.2) ile gerçek dünya veri seti sonuçları aynı beş meta-sezgisel algoritma, aynı  $F(x)$  (Bölüm 3.4), aynı referans çizgisi politikası (Bölüm 3.4.4) ve aynı arama limitleri altında yürütülmektedir. Tek yapısal fark, görev havuzunun kaynağıdır: Poisson( $\lambda$ ) senaryo üretici (Bölüm 3.1) ile beş gerçek dünya veri setinden türetilen havuzlar (Bölüm 2.6.2; Bölüm 3.5.3). Bu ortak zemin AS3 hipotezinin test edilmesini sağlamaktadır (Guo vd. 2019; Hadary vd. 2020; Mach vd. 2017).

Çapraz kıyas okunurken üç noktaya dikkat edilmelidir. Birincisi, Poisson senaryolarında birincil ölçütler gecikme ve kaynak tüketimidir (Bölüm 3.4.2). Veri seti ekseninde yanıt süresi milisaniye olarak hesaplanmakta, kaynak yüzdesi sonuç çıktılarında yer almamaktadır (Bölüm 4.1.5). Mutlak milisaniye değerleri doğrudan toplanamaz. Göreceli sıralama ve profil–algoritma eşlemesi esas alınmalıdır (Mao vd. 2017). İkincisi, tek bir veri setinde evrensel kazanan bulunmamaktadır (AS1). Üçüncüsü, tekli koşullar keşifsel kanıt niteliğindedir. Çoklu test koşullarının  $n=10$  tekrarlı özeti (Çizelge 4.7) AS1'i teyit etmekte, özellikle AZURE\_PUBLIC'te başlangıç değerleri duyarlılığının tekli koşudaki dar banttı sapabildiği görülmektedir (Deb 2002; Rao 2009).

ACO ve ABC için raporlanan üst tr aralığı, MLPerf ve hibrit havuzlardaki büyük task\_size kayıtlarından kaynaklanmaktadır; bu eğilim, orta kaynaklı Poisson profillerinde ACO'nun göreceli zayıflığıyla uyumludur (Dorigo vd. 2006). PSO ve GWO'nun hem yüksek yoğunluklu senaryolarda hem çoklu veride düşük tr bandına yakın kalması AS2'yi desteklemektedir (Bölüm 4.1.16; Mach vd. 2017).

Çapraz kıyasın özeti şu şekilde ifade edilebilir. İlk olarak, iki eksen AS3 altında karşılaştırılabilir sonuç üretmiştir. İkinci olarak, AS1 desteklenmektedir: ne Poisson

ortalaması ne de tek bir veri seti tüm algoritmalar için sürekli lider seçmemektedir. Üçüncü olarak, AS2 kapsamında değerlendirmede profil sınıfına göre aday algoritma seçimi önerilmektedir. Dördüncü olarak, tekli koşullar hipotez üretimi için yeterli ön bulgu sağlamaktadır; çoklu test koşullarının n=10 tekrarlı özeti (Çizelge 4.7) PSO/GWO önceliğini çoğu veri setinde desteklemektedir (Deb 2002).

#### 4.1.13 Çapraz bulgular ve AS1–AS4 bağlantısı

Bu alt bölümde, yukarıdaki çapraz kıyas sonuçları dört araştırma sorusu (AS1–AS4) ile eşleştirilmektedir. Amaç, sayısal bulguların tez sorularına hangi yanıtı verdiğini ayrı ayrı ve okunabilir biçimde göstermektir.

AS1 — Tek bir algoritma her profilde lider değildir. Örneğin, veri seti ekseninde MLPerf'te GWO (12,65 ms), 5G'de PSO (11,77 ms), Alibaba'da GWO/PSO/ABC eşit (12,28 ms) elde edilmiştir (Çizelge 4.2). Poisson ekseninde yüksek yoğunluklu kentselde PSO/GWO, tarımsal ağda GA/ABC/ACO en düşük gecikme değerlerini üretmiştir (Şekil 4.3–4.4 ve 4.13–4.14). Birleşik modlarda liderlik GWO veya ABC'ye kaymıştır (Çizelge 4.5). On beş senaryo ortalamasında GA/ABC daha düşük gecikme değerleri üretirken tekil profillerde PSO/GWO baskın olabilmektedir (Şekil 4.31–4.32; Mach vd. 2017).

AS2 — Üstünlükler algoritma operatörleriyle uyumludur. Örneğin, MLPerf'te PSO/GWO kompakt görev boyutuna ( $\approx 10$ ) yakınsamakta; ACO ağır kuyruk (task\_size $\approx 144$ ) üretmektedir (Çizelge 4.4). Poisson ekseninde sınırlı internet kırsalda GWO/PSO zayıf, GA/ABC/ACO güçlü kalmıştır (Şekil 4.11–4.12). PSO/GWO yüksek yoğunlukta  $\sim 11$ –13 ms bandında kalmaktadır (agresif keşif; Kennedy vd. 1995; Mirjalili vd. 2014). GA/ABC düşük kaynak profillerinde avantajlıdır (Goldberg 1989; Karaboga vd. 2007). ACO geniş görev boyutunda yavaşlamaktadır (Dorigo vd. 2006; Reddi vd. 2020).

AS3 — İki deneysel eksen karşılaştırılabilir sonuç üretmiştir. Ortak zemin: aynı  $F(x)$ , yerel/uç değerlendirme, popülasyon 100, iterasyon 100 (Bölüm 3.5.3). Örneğin, yüksek

yoğunluklu kentsel Poisson senaryosunda PSO/GWO üstünlüğü, MLPerf çıkarım profilinde PSO/GWO bandıyla tutarlıdır. Poisson ortalamasında GA/ABC daha düşük gecikme değerleri üretirken 5G veri setinde PSO en düşük ortalama değeri vermiştir; fark, ortalama profil ile gerçek kuyruk davranışı arasındaki ayırmadan kaynaklanmaktadır (Guo vd. 2019; Hadary vd. 2020). Birleşik modlar sıralamayı değiştirmektedir (Çizelge 4.5/4.9).

AS4 — Profil tanımlandığında algoritma seçilebilmektedir. Karar akışı şu aşamalardan oluşmaktadır: (1) profil sınıfının tanımlanması (Poisson  $\lambda$  merkezi veya veri seti kökeni); (2) birincil metriğin seçilmesi (gecikme ve kaynak veya  $tr$ ); (3) tekli koşu ile yönün belirlenmesi (Çizelge 4.2/4.5); (4) çoklu test ile doğrulanması (Çizelge 4.7/4.9); (5) istatistiksel test ile iddianın sınırlandırılması (Çizelge 4.8). Poisson ekseninde yüksek yoğunlukta PSO/GWO, düşük kaynakta GA/ABC öncelikli görülmektedir; gerçek dünya veri setlerinde ise profil kökenine göre Bölüm 4.1.3–4.1.4 sonuçları esas alınmaktadır. Çizelge 4.7 ve 4.9 tekrarlı sonuçları doğrulamakta veya sınırlandırmaktadır; Çizelge 4.8 istatistiksel iddiayı daraltmaktadır (Bölüm 4.1.16; Deb 2002).

#### 4.1.14 Literatür uyum değerlendirmesi (Veri Seti Kökenli Yığın)

Üç deney modu birlikte değerlendirildiğinde yığın, veri seti güdümlü MEC literatürüyle şu noktalarda hizalanmaktadır. (1) Gerçek veri önceliği: beş veri seti (Alibaba, Azure, GeoLife, 5G, MLPerf) dataset-only modda bağımsız değerlendirilmiştir (Guo vd. 2019; Hadary vd. 2020). (2) Ölçek korunumu: dataset-only modda min–max kalibrasyon kapalı bırakılmıştır. (3) Simülatör güvenliği: Azure profilinde sınır kırpması aşırı değerleri engellemiştir (Hadary vd. 2020). (4) Dual-path değerlendirme: 5G profilinde yerel/uç seçimi kanal koşuluna göre değişmiştir (Mao vd. 2017; Mach vd. 2017). (5) İzlenebilir raporlama: koşu parametreleri Çizelge 4.6'da özetlenmiştir (Rao 2009).

Bu bölümde raporlanan tek koşuluk ve tekrarlı deneyler, birinci–ikinci aşama deneysel kanıt sunmaktadır. Hibrit ve adaptör modları entegrasyon politikasının etkisini ayrıca test etmiştir (AS3). Gelecek çalışmalarda zamansal veri bölme stratejileri, Poisson ekseninde  $n \geq 10$  tekrarlı koşu tamamlanması ve çoklu uç sunucu senaryoları protokolü

güçlendirebilecek başlıklar olarak değerlendirilmektedir (Guo vd. 2019; Taleb vd. 2017; Deb 2002).

#### 4.1.15 Bulguların sentezi ve algoritma seçim çerçevesi

Bölüm 4.1 (on beş Poisson senaryosu) ve Bölüm 4.1.3 (beş gerçek dünya veri seti ve birleşik modlar) birlikte okunması gerekmektedir. Ortak bulgu: tek evrensel kazanan algoritma yoktur. Performans senaryo kaynak profili, ağ kapasitesi ve dinamiklik derecesine bağlıdır. Tekli koşullarda dataset-only modda en düşük yanıt süreleri 11,77 ms (PSO, 5G) ile 223,19 ms (GA, MLPerf) arasında değişmiştir. Birleşik modda GWO 11,66 ms ve ABC 11,62 ms ile alt banda inmiştir. ACO'nun MLPerf ve hibrit havuzlarda 1163–1220 ms mertebesine çıkması, feromon tabanlı aramanın geniş görevi büyüklüğü bölgelerinde takılabildiğini göstermektedir (Dorigo vd. 2006; Reddi vd. 2020). Bu sonuç AS4'e olumlu yanıt vermektedir (Mach vd. 2017).

Üç değerlendirme katmanı birlikte değerlendirildiğinde ortak üç bulgu çıkmaktadır. Birincisi, tek evrensel kazanan algoritma bulunmamaktadır (AS1). İkincisi, üstünlükler operatör mekanizmalarıyla uyumludur: agresif keşif (PSO/GWO) yüksek kaynak profillerinde, limit/elitizm (GA/ABC) düşük kaynak profillerinde, feromon araması (ACO) orta profillerde veya geniş görev boyutunda zayıflamaktadır (AS2). Üçüncüsü, entegrasyon politikası (dataset-only, hibrit, adaptör) sıralamayı değiştirebilmektedir (AS3).

Karar çerçevesi şu sırayla uygulanması önerilmektedir: (1) profil sınıfının tanımlanması (Poisson  $\lambda$  merkezi veya veri seti kökeni); (2) birincil metriğin seçilmesi (gecikme ve kaynak veya tr); (3) tekli koşu ile yönün belirlenmesi (Çizelge 4.2/4.5); (4) çoklu test ile doğrulanması (Çizelge 4.7/4.9); (5) istatistiksel test ile iddianın sınırlandırılması (Çizelge 4.8).

## 4.2 Tartışma

Bu çalışma, farklı MEC senaryolarında optimizasyon algoritmalarının performansları sistematik biçimde değerlendirilmiştir. Sonuçlar, tek bir algoritmanın tüm senaryolarda en iyi performansı gösteremeyeceğini, bunun yerine senaryo özelliklerine göre algoritma seçiminin yapılması gerektiğini ortaya koymaktadır. Bu çalışmada elde edilen bulgular algoritmaların karakteristik özelliklerinin sonuçlara doğrudan etkilediği gözlenmektedir. Özellikle hibrit yaklaşımlar ve senaryo-bazlı parametre optimizasyonlar gelecek çalışma olarak değerlendirilmektedir.

### 4.2.1 Araştırma sorularına yanıtların özeti

Bölüm 1.2'deki problem tanımına yanıt olarak: AS1 — Tek üstün algoritma yoktur. Hem Poisson senaryolarında (Bölüm 4.1.2) hem gerçek dünya veri setlerinde (Bölüm 4.1.3) profil değişince lider değişmiştir. AS2 — Üstünlükler operatör özellikleriyle uyumludur:

- Elitizm/limit (GA, ABC),
- Agresif keşif (PSO, GWO),
- Feromon maliyeti (ACO).

AS3 — Aynı  $F(x)$  altında Poisson senaryoları ve gerçek dünya veri setleri karşılaştırılabilir sıralama üretmiştir. Referans çizgisi iyileşmesi her iki ekseninde de görülmüştür (Guo vd. 2019; Hadary vd. 2020). AS4 — Çizelge 4.2 ve Çizelge 4.7'deki bulgular seçim çerçevesini operasyonel hale getirmektedir.

### 4.2.2 Genel sonuç ve seçim çerçevesi önerisi

Tez, “tek en iyi meta-sezgisel” arayışından çok, “bağlam tanımlandığında hangi ailenin tercih edileceği” sorusuna yanıt vermektedir. Önerilen değerlendirme akışı şu şekilde özetlenebilir:

- Senaryo veya veri seti profilini Çizelge 3.7 ile sınıflandır.
- QoS önceliği gecikme ise ve kaynak bol ise PSO veya GWO dene.
- Kaynak kısıtlı veya enerji hassas ise GA veya ABC dene.
- Orta kaynak ve feromon maliyeti tolere edilemiyorsa ACO'yu düşük öncelikli tut.
- Felaket/dinamik profilde GA'yı önceliklendir.
- $n \geq 10$  tekrar koşu ile ortalama $\pm$ std doğrula (Bölüm 4.1.9).

Bu akış, MEC operatörlerine doğrudan politika değil karar destek çerçevesi sunmaktadır. Deneyisel kanıt iki ekseninde okunmuştur: on beş Poisson senaryosu (Bölüm 3.1) ve beş gerçek dünya veri seti (Bölüm 3.5.3, 4.1.3). Tekli koşullarda veri seti başına 11,62–223,19 ms aralığında değişen yanıt süreleri gözlenmiş ve tek evrensel kazanan bulunmamıştır (AS1). Metrikler simülatör formüllerinden türetilmiştir; saha ölçümü ile çapraz doğrulama gelecek çalışmanın konusudur (Bölüm 4.2.6; Mao vd. 2017; Mach vd. 2017).

#### 4.2.3 MEC'de meta-sezgisel kullanımın sağladığı faydaların sentezi

Deneyisel bulgular, MEC özelinde meta-sezgisel kullanımın dört somut faydasını desteklemektedir:

- (1) Bağlama duyarlı performans: Tek algoritma tüm senaryolarda üstün değildir. Bu, MEC'de statik tek politikanın yetersizliğini doğrular (Cao vd. 2019) (Mach vd. 2017). Yüksek yoğunluklu kentsel ve etkinlik trafiğinde PSO ve GWO gecikmede öne çıkar. Kırsal ve tarımsal düşük kaynak profillerinde GA ve ABC kaynak metriğinde daha dengeli kalmaktadır (Kennedy vd. 1995) (Goldberg 1989) (Karaboga vd. 2007).
- (2) Operatörle uyumlu trade-off: PSO/GWO agresif keşifle kaynak harcayarak gecikmeyi düşürür; QoS öncelikli uygulamalarda kabul edilebilir bir değiş tokuştur (Mirjalili vd. 2014). GA/ABC düşük kaynaklı ortamlarda aşırı uç yüklemesini sınırlandırarak sürdürülebilirliğe katkı sağlamaktadır (Goldberg 1989).
- (3) Referans çizgisine göre iyileşme: Tüm optimizier'ların tümü yerel veya tümü uç referanslarına göre  $F(x)$  iyileştirmesi, meta-sezgisel aramanın sabit kuraldan daha iyi olduğunu göstermektedir; karar değişkenlerinin birlikte optimize edilmesi gerektiğini

vurgulamaktadır (Mao vd. 2017).

(4) Literatürle tutarlı operatör açıklaması: ACO'nun orta kaynak profillerinde feromon güncelleme maliyeti (Dorigo vd. 2006); GA'nın dinamik felaket senaryosunda popülasyon çeşitliliği (Goldberg 1989); sonuçlar algoritma ailelerinin MEC'de neden ve nasıl seçileceğine dair kuramsal beklentiyle örtüşmektedir (Shakarami vd. 2020).

Özetle, MEC'de meta-sezgisel yöntemler evrensel tek çözüm değil; profil tanımlandığında gecikme, kaynak ve uyarlanabilirlik hedeflerini dengeleyen uygulanabilir karar araçlarıdır (Taleb vd. 2017) (Rodrigues vd. 2020).

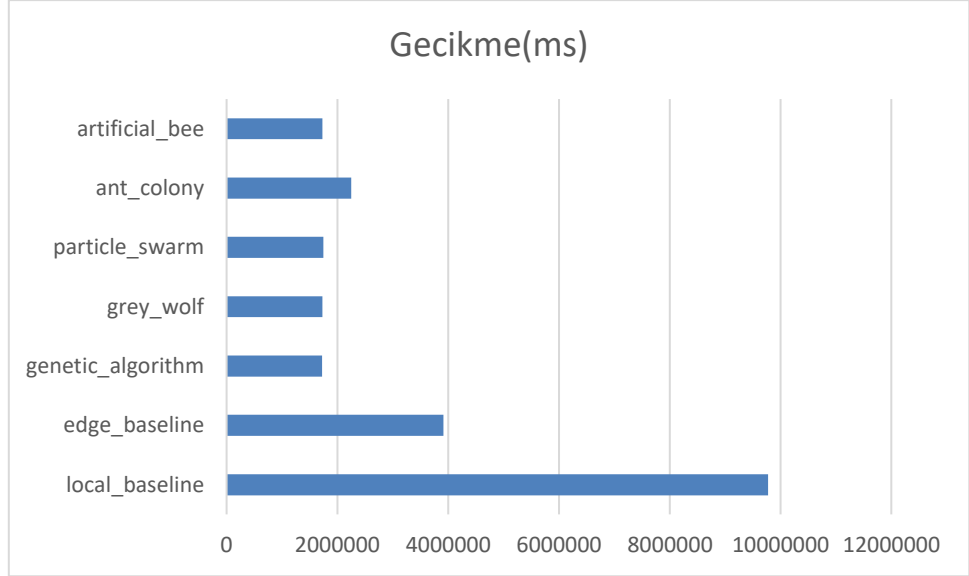
#### 4.2.4 Poisson senaryo sonuçlarının değerlendirilmesi

Bölüm 4.1'deki on beş senaryo, gerçek dünyadaki isimlerle (kırsal, akıllı şehir, felaket kurtarma vb.) etiketlenmiştir; her biri farklı  $\lambda$  profiliyle aynı  $F(x)$  altında test edilmiştir. Okurken şu sırayı öneririz: (1) İlgili Şekil 4.x gecikme ve kaynak eğrilerine bakın — hangi algoritma çizgisi aşağıda? (2) Kısa özet paragrafları okuyun. (3) Senaryo yorum paragrafını (Bölüm 4.1.2) AS2 bağlantısı için kullanın. (4) Ortalama Şekil 4.31–4.32 ile genel eğilimi, Şekil 4.35–4.36 ile dinamik/statik ayrımını kontrol edin. Dinamik ve statik senaryo gruplamaları sırasıyla Şekil 4.33–4.34 ve Şekil 4.35–4.36'da gösterilmektedir. Referans çizgisi (tümü yerel / tümü uç) eğrileri, optimizasyonsuz referanstır: optimizasyon'un bu eğrilerin altına inmesi, meta-sezgisel aramanın anlamlı olduğunu göstermektedir (Bölüm 3.5.2). En iyi %20 raporlama kuralı, PSO/GWO'nun keşif sırasında ortalama skorunu düşürmesini engellemek içindir; yalnızca en iyi bireyler kıyaslanmaktadır (Kennedy vd. 1995).

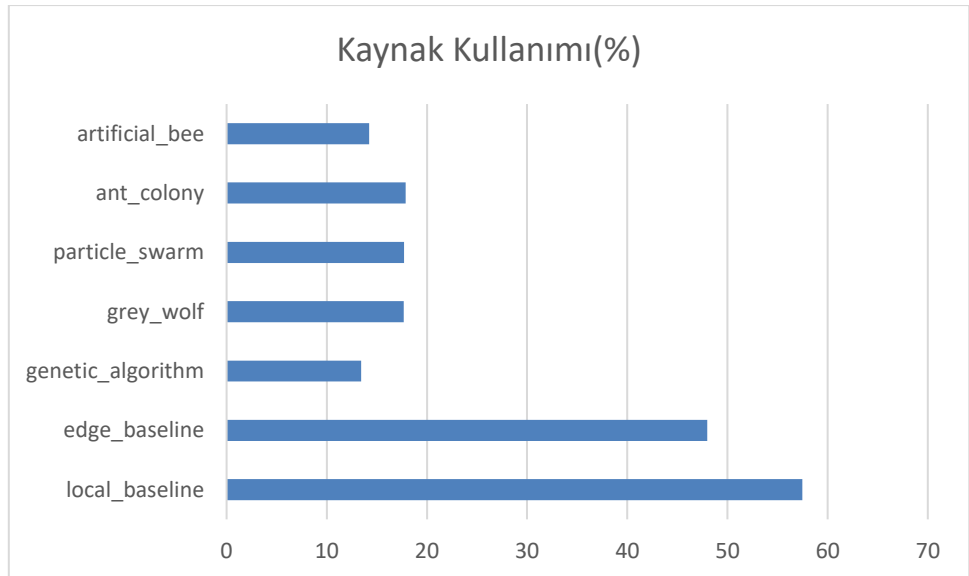
Tek senaryoda “kazanan” çıkmaması tez iddiasıyla çelişmez; tam tersine AS1'i desteklemektedir: MEC'de algoritma seçimi profil ile birlikte düşünülmelidir (Cao vd. 2019) (Mach vd. 2017). Mutlak ms değerleri simülatör iç tutarlılığı içindir; senaryolar arası sıralama ve aile eğilimleri asıl kanıttır (Mao vd. 2017).

Dinamik ve statik senaryo grupları Şekil 4.33–4.36'da özetlenmiştir. Simülasyon sonuçları, farklı senaryolarda iyileştirici (optimizasyon) algoritmalarının performanslarının önemli ölçüde değiştiğini göstermektedir. Özellikle yüksek yoğunluklu senaryolarda

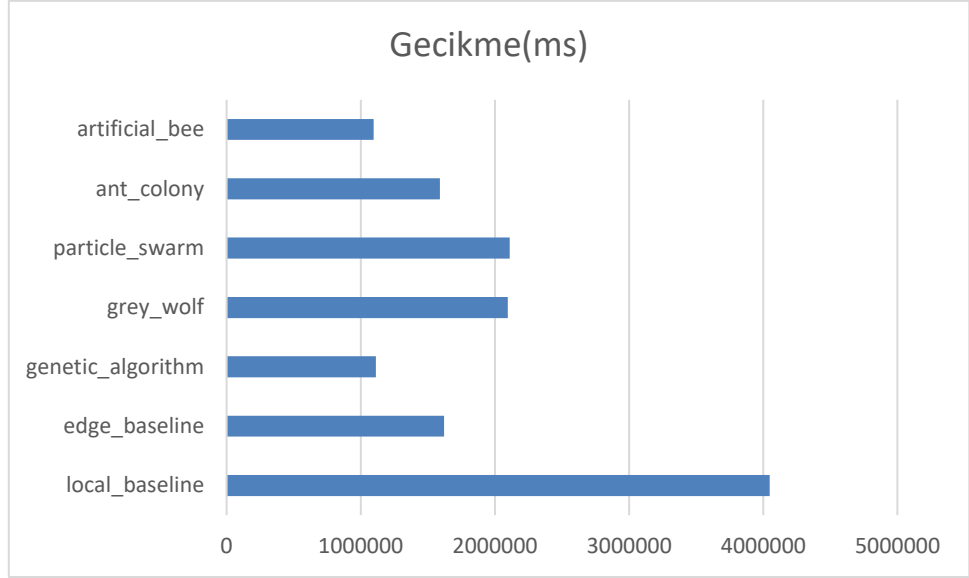
(Yüksek Yoğunluklu Kentsel Alan, Etkinlik Kaynaklı Yüksek Trafik gibi) ve düşük kaynaklı senaryolarda (Sınırlı İnternet Erişimi Olan Kırsal Bölge, Tarımsal Ağ İhtiyaçları gibi) algoritmaların davranışları belirgin farklılıklar göstermektedir. Bu da bizlere değişik senaryolar için uygun çözümler üretmenin önemini göstermektedir. Aşağıda dinamik ve statik senaryolara göre gruplanmış deney sonuçları görülmektedir.



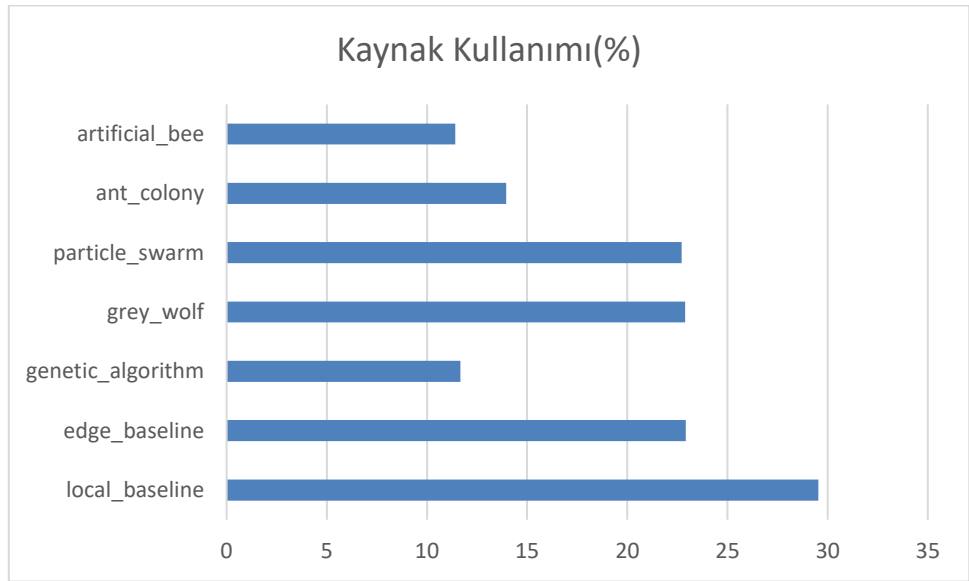
Şekil 4.33 Dinamik senaryolar için gecikme



Şekil 4.34 Dinamik senaryolar için kaynak kullanımı



Şekil 4.35 Statik senaryolar için gecikme



Şekil 4.36 Statik senaryolar için kaynak kullanımı

GWO ve PSO genel olarak birbirine yakın performanslar göstermekte ve yüksek kaynaklı ve yoğunluklu senaryolarda daha başarılı olmaktadır. Kaynak tüketimini artırarak daha kısa cevap süresi sağlamaktadırlar. Düşük kaynak içeren senaryolarda ise oldukça başarısız sonuçlar üretmişlerdir. Bu anlaşılabilir bir durumdur çünkü bu iki algoritma daha yüksek risk alan agresif keşif yapan algoritmalarıdır. Elde edilen sonuçlar literatür ile uyumludur. (Mirjalili vd. 2014) (Kennedy vd. 1995) Aldıkları risk düşük kaynak içeren senaryolarda başarıya ulaşmamakla birlikte yüksek kaynak içeren senaryolarda

olumlu sonuçlar vermişlerdir. Buna karşın GA ve ABC düşük kaynak içeren senaryolarda daha etkili olmuşlardır. Ayrıca kaynak kullanımı metriğinde daha dengeli sonuçlar vermişlerdir. GA ve ABC için elde edilen sonuçlar da literatürle uyumludur. (Goldberg 1989) (Karaboga vd. 2007) ACO ise düşük ve yüksek kaynak senaryolarında ikili grupların arasında, tüm algoritmaların benzer sonuç ürettiği orta kaynak senaryolarında diğer algoritmalara göre daha kötü performans göstermiştir. Dorigo vd. 2006'daki çalışmada ACO'nun orta kaynaklı problemlerde diğer algoritmalara göre daha yavaş yakınsadığını, çünkü feromon güncelleme mekanizmasının zaman aldığını belirtmiştir. Dinamik ortam senaryolarında tüm algoritmalar performans düşüşü yaşamaktadırlar. Ancak GA'nın daha hızlı uyum sağladığı görülmüştür. Bu denge senaryo gereksinimlerine değişkenlik göstermektedir. Bir diğer bulgu ise tüm optimizasyon algoritmaları referans çizgileri (referans çizgisi)ne göre önemli iyileştirme sağlamasıdır.

Özetle, düşük kaynaklı senaryolarda GA veya ABC tercih edilmelidir. Yüksek yoğunluklu senaryolarda PSO veya GWO kullanılmalıdır. GA'nın düşük kaynak tüketimi ile PSO/GWO'nun hızlı yanıt süresi avantajları birleştirilebilir ve senaryo tipine göre dinamik algoritma seçimi yapılabilir. Özellikle, GA'nın elitizm ve çeşitlilik koruma mekanizmaları sayesinde düşük kaynaklı ortamlarda daha dengeli performans göstermesi (Goldberg 1989), ABC'nin ise gözcü ve izci arı davranışları ile kaynak kısıtlamalarına karşı daha dayanıklı olması (Karaboga vd. 2007), bu algoritmaların düşük kaynaklı senaryolarda tercih edilmesinin temel nedenlerini oluşturmaktadır. Diğer yandan, PSO'nun sosyal ve bireysel öğrenme mekanizmaları ile GWO'nun liderlik hiyerarşisi ve Levy flight özellikleri, yüksek yoğunluklu ve karmaşık senaryolarda daha agresif keşif yaparak hızlı yakınsama sağlamaktadır (Kennedy vd. 1995; Mirjalili vd. 2014). Bu algoritmaların yüksek kaynak tüketimi karşılığında elde ettikleri performans artışı, özellikle QoS gereksinimlerinin kritik olduğu durumlarda kabul edilebilir bir trade-off olarak değerlendirilebilir. Bu yaklaşım, MEC sistemlerinin farklı ortam koşullarına dinamik olarak uyum sağlamasını ve optimal performans elde etmesini sağlayacaktır. Elde edilen bulgular, MEC sistemlerinde algoritma seçimi konusunda sistematik bir yaklaşım geliştirilmesi gerektiğini göstermektedir ve gelecekteki MEC sistemleri için algoritma seçimi ve optimizasyon stratejileri konusunda önemli rehberlik sağlamaktadır.

#### 4.2.5 Gerçek Dünya veri seti sonuçlarının değerlendirilmesi

Veri seti deneylerinin bulguları bu bölümde ele alınacaktır. Değerlendirme üç metodolojik ilkeye dayanmaktadır. Birincisi, tüm algoritmalar aynı amaç fonksiyonu  $F(x)$ , aynı arama bütçesi (popülasyon 100, iterasyon 100) ve aynı yerel/uç değerlendirme protokolü altında kıyaslanmıştır. Dolayısıyla gözlenen farklar veri üretim hattından değil, arama operatörlerinin görev uzayına uyumundan kaynaklanmaktadır (Bölüm 4.1.5). İkincisi, raporlanan yanıt süreleri simülatör ölçeğindedir. Mutlak milisaniye değerleri veri setleri arasında toplanamaz. Göreli sıralama ve yerleşim kararı esas alınmalıdır (Mao vd. 2017). Üçüncüsü, tekli koşullar keşifsel kanıt üretir. Çoklu test ortalamaları bu kanıtı teyit eder veya sınırlar. Özellikle Azure profilinde bu ayrım belirleyicidir (Deb 2002; Rao 2009; Hadary vd. 2020).

Çizelge 4.2 ve Çizelge 4.7 birlikte okunduğunda AS1 desteklenmektedir: tek bir meta-sezgisel tüm veri setlerinde sürekli lider değildir. MLPerf profilinde GWO (12,65 ms) ve PSO (12,82 ms) dar bantta kalırken GA (223,19 ms) ile ACO/ABC (1163 ms) belirgin ağır kuyruk üretmiştir. Bu dağılım, kompakt görev boyutlu çıkarım yüklerinde sürü tabanlı keşfin GA elitizmi ve ACO feromon maliyetine göre üstün kaldığını göstermektedir (Reddi vd. 2020). 5G Monarch veri setinde PSO (11,77 ms) hem tekli hem çoklu testte lider konumunu korumuştur. Bu bulgu, saha ölçümü kökenli kanal profillerinde sosyal öğrenme mekanizmasının pratik anlamda anlamlı bir avantaj sağlayabileceğini düşündürmektedir (Liu vd. 2023). GeoLife mobilite veri setinde GWO önceliklidir. Alibaba küme iş yükünde ise üç algoritma (PSO, GWO, ABC) aynı 12,28 ms bandına inerken tüm algoritmalar uç yürütmeyi seçmiştir. Bu örüntü, veri merkezi kaynak profillerinde aktarım maliyetinin baskın olduğu literatür bulgularıyla uyumludur (Guo vd. 2019; Zheng vd. 2010).

Yanıt süresi ayrışmasının kaynağı, Çizelge 4.4'teki bileşen analiziyle netleşmektedir: algoritmalar arasındaki farkın büyük bölümü yürütme süresinden (te) gelmektedir. İletim gecikmesi (td) görece sabit kalmaktadır. Dolayısıyla tartışmanın odak noktası, hangi arama operatörünün hangi görev boyutu dağılımına ve yerleşim seçeneğine daha hızlı yakınsadığıdır. MLPerf ve 5G profillerinde yerel önceliği, GeoLife'da yerel işleme

eğilimi, Alibaba'da ise tam uç yerleşimi bu mantıkla okunması gerekmektedir: meta-sezgisel seçim yalnızca gecikme minimizasyonu değil, aynı zamanda aktarım–yürütme dengesinin profil tarafından belirlenmesidir (You vd. 2017; Guo vd. 2019).

Azure profili, tekli ve çoklu test değerlendirme katmanları arasındaki gerilimi belirgin biçimde göstermektedir. Tekli koşuda tüm algoritmalar 12,28–17,25 ms dar bandındayken. Çoklu testte yalnızca GWO  $14,19 \pm 2,50$  ms ile kararlı kalırken GA ( $8143 \pm 13\,168$  ms), ACO ( $48\,175 \pm 67\,804$  ms), PSO ( $1638 \pm 2104$  ms) ve ABC ( $24\,423 \pm 46\,850$  ms) yüksek ortalama ve standart sapma üretmiştir. Bu bulgu, sınır kırpmasının (TARGET\_BOUNDS) tek koşuda gözlenen dar bandı garanti etmediğini ve VM packing kökenli profillerde başlatma kümesine bağlı sonuç değişkenliğinin yanlış genelleme riski oluşturduğunu göstermektedir. Bilimsel çıkarım şudur: Azure benzeri profillerde algoritma seçimi yalnızca tekli koşuya değil, çoklu test ortalamasına dayandırılması gerekmektedir (Hadary vd. 2020; Ben-David vd. 2010). MLPerf izinde de çoklu testte GA ve ACO ortalamalarının yükselmesi (309 ms ve 1335 ms) aynı duyarlılığın daha kontrollü bir biçimde tekrarlandığını düşündürmektedir.

Birleşik mod deneyleri (Çizelge 4.5, Çizelge 4.9) ana bulgu yönünü değiştirmez. Tamamlayıcı bulgu sunmaktadır. Hibrit havuzda GWO çoklu testte  $12,80 \pm 0,74$  ms ile kararlı kalırken ACO yüzlerce–bin ms bandına çıkmıştır. Bu, karışık görev havuzlarında feromon tabanlı aramanın zayıflayabileceğini destekler (Dorigo vd. 2006). Global kalibrasyonda tekli koşuda ABC en düşük gecikmeyi üretmiş olsa da tekrarlarda GWO en düşük ortalama değeri vermiştir. Bu modlar, veri setlerinin farklı birleştirme politikaları altında nasıl davrandığını test etmektedir. Ancak operatör düzeyindeki seçim önerisi dataset-only bulgularına dayanmalıdır (Sohaib vd. 2024).

Gerçek veri eksenine özgü bulguların Poisson tartışmasıyla nasıl bir araya geldiği ve bu birleşik okumanın araştırma sorularına hangi yanıt verdiği, aşağıdaki alt bölümde ele alınmaktadır.

#### 4.2.6 Deneyisel eksenlerin birlikte değerlendirilmesi

Bu bölümde, Poisson tabanlı senaryoları ile veri setleriyle yapılan test koşullarının birlikte değerlendirmesi yapılacaktır.

İki eksenin birlikte değerlendirilebilmesinin önkoşulu, karar probleminin özdeş kalmasıdır. Her iki kanalda da aynı standart öznitelik şeması, aynı amaç fonksiyonu  $F(x)$ , aynı yerel/uç değerlendirme protokolü ve aynı arama limiti kullanılmıştır (Bölüm 4.1.5). Fark, optimizasyon probleminden değil, görev havuzunun nasıl üretildiğinden kaynaklanmaktadır: Poisson ekseninde  $\lambda$  merkezleriyle etiketlenmiş on beş profil; gerçek kayıt ekseninde ise MLPerf, 5G Monarch, GeoLife, Alibaba ve Azure kayıtlarının adaptör hattından geçirilmiş hali (Bölüm 3.5.4). Dolayısıyla tartışma, "hangi algoritma her koşulda üstündür?" sorusunu değil, "iş yükü profili tanımlandığında hangi ailenin hangi değerlendirme katmanında öne çıktığı" sorusunu yanıtlamaktadır.

AS3 ve AS4, bu birlikte okumanın tez düzeyindeki karşılığıdır. AS3 kapsamında iki eksen, aynı simülatör ve aynı raporlama politikası altında karşılaştırılabilir sıralama üretmiştir; yani "Poisson'da iyi" ile "gerçek veride iyi" arasında mekanik bir eşitlik yoktur, ancak profil eşlemesi yapıldığında yön uyumu sıklıkla görülmektedir. Örneğin yüksek yoğunluklu kentsel ve akıllı şehir senaryolarında PSO/GWO önceliği, MLPerf ve 5G veri setlerinde ~11–13 ms bandına yakınsamayla niteliksel olarak örtüşür (Reddi vd. 2020; Liu vd. 2023). AS1 her iki ekseninde de desteklenmektedir: ne Poisson ortalaması ne de tek bir veri seti sürekli lider seçmemektedir. AS2, entegre düzeyde şöyle yorumlanır: PSO/GWO agresif keşif ile QoS profillerinde. GA/ABC ise kaynak dengesi ve limit mekanizmaları ile düşük kaynak Poisson profillerinde öne çıkar. ACO feromon maliyeti nedeniyle büyük task\_size ve karışık havuzlarda dikkatle değerlendirilmelidir. AS4 için önerilen değerlendirme akışı şu şekilde özetlenebilir: (i) profil sınıfına göre aday algoritma kümesini belirle. (ii) mümkünse Çizelge 4.7–4.9 çoklu test ortalamalarıyla teyit et. (iii) Azure benzeri profillerde tekli koşu sonucunu tek başına kullanma (Ben-David vd. 2010).

Son olarak, entegre değerlendirmenin sınırları açık tutulmalıdır. Bu tez, Poisson

senaryoları ile veri setlerinin aynı  $F(x)$  altında "aynı sayıları" üreteceğini iddia etmez. İddia, profil tanımlandığında algoritma seçiminin sistematik hale getirilebileceğidir. İki eksen birlikte okunduğunda ortaya çıkan en güçlü bulgu, MEC'de meta-sezgisel seçimin bağlamdan bağımsız bir sıralama oyunu olmadığıdır: araştırmacı, önce iş yükünün kaynağını (sentetik profil mi, gerçek veri mi), ardından değerlendirme katmanını (tekli mi, çoklu mu) tanımlamalı; ancak o zaman PSO, GWO, GA, ABC veya ACO ailesi arasında gerekçeli tercih yapılabilir. Bu çerçevede, Bölüm 4.2.2'de özetlenen karar destek yaklaşımının deneysel temelini oluşturur ve sonraki bölümlerde ele alınan erken yayın bulguları ile sınırlamalar bağlamında konumlandırılmalıdır.

#### 4.2.7 Önceki yayın bulguları ile karşılaştırma

Bu tez çalışmasının erken aşamasında, üç sezgisel algoritma (GA, PSO, GWO) ve iki referans politika (Tümü Yerel, Tümü Uç) ile 500 batch'lik simülasyon protokolü uygulanmış; bulgular makale olarak yayınlanmıştır. Bu tezde ACO ve ABC eklenmiş, on beş Poisson senaryosu ve beş gerçek dünya veri seti üzerinde ikinci deneysel eksen tanımlanmıştır (Holland 1975) (Kennedy vd. 1995) (Mirjalili vd. 2014) (Dorigo vd. 2006) (Karaboga vd. 2007). Erken makale koşusunda elde edilen temsili sonuçlar Çizelge 4.10'te özetlenmiştir. Tümü Yerel referans çizgisi gecikme 7200,602 ms ve kaynak %43,31 ile en yüksek gecikme ve kaynak değerlerini vermiş; Tümü Uç 2880,746 ms ve %35,41 ile referans oluşturmuştur. GA 2184,918 ms / %16,99, PSO 1801,364 ms / %19,14 ve GWO 1727,074 ms / %18,71 değerleriyle her iki referans çizgisini anlamlı biçimde geçmiştir.

Makaledeki karşılaştırma kuralı Bölüm 3.5.2'deki elit karşılaştırma politikasıyla uyumlu görülmektedir. GWO ve PSO gecikmede birbirine yakın ve en düşük banda yakınsarken, GA kaynak tüketiminde görece avantaj göstermiştir. Bu tezdeki Poisson senaryoları (Bölüm 4.1) ve gerçek dünya veri seti sonuçları (Bölüm 4.1.3) önceki yayın bulgularıyla niteliksel olarak uyumludur. Tek evrensel kazanan olmaması AS1 iddiasıyla tutarlıdır. Metrikler her iki yayında da simülasyon-türetimlidir. Saha ölçümü iddiası taşınmamaktadır (Mach vd. 2017).

Çizelge 4.10 Erken makale koşusu: temsili gecikme ve kaynak sonuçları

| Algoritma / Politika   | Gecikme (ms) | Kaynak tüketimi (%) |
|------------------------|--------------|---------------------|
| Tümü Yerel (baseline)  | 7 200,602    | 43,31               |
| Tümü Uç (baseline)     | 2 880,746    | 35,41               |
| Genetik Algoritma (GA) | 2 184,918    | 16,99               |
| Parçacık Sürüsü (PSO)  | 1 801,364    | 19,14               |
| Gri Kurt (GWO)         | 1 727,074    | 18,71               |

#### 4.2.8 Çalışmanın sınırlamaları ve gelecek çalışmalar

Sınırlamalar şeffaf biçimde şöyle özetlenmektedir: (S1) Metrikler saha ölçümü değil, Bölüm 3.4 formüllerinin çıktısıdır. Mutlak ms değerleri yalnızca simülatör iç tutarlılığı içindir. (S2) Tek görev–tek karar modeli benimsenmiştir. Çoklu eşzamanlı görev, kısmi aktarım ve çoklu uç sunucu topolojisi kapsam dışıdır (EdgeCloudSim/iFogSim tam topoloji modelleri, Bölüm 2.6.3). (S3) Hiperparametreler grid araması ile sabitlenmiştir. Senaryo başına otomatik ayar yapılmamıştır. (S4) Azure+GA birleşiminde, simülatör sınır kırpması kapalıyken (clamp\_to\_simulator\_bounds=False) yapılan erken tekli koşuda ölçek patlaması gözlenmiştir. Kırpma açıldığında tekli koşuda tüm algoritmalar dar bantta kalmıştır. Buna karşılık çoklu test koşularında AZURE\_PUBLIC profilinde GA, ACO, ABC ve PSO yüksek ortalama ve standart sapma üretmiştir. Kırpma tek koşu için güvenlik sağlasa da başlangıç değerlerine bağlı aşırı değerleri tamamen ortadan kaldırmamaktadır (Bölüm 4.1.9, Çizelge 4.7; Hadary vd. 2020; Ben-David vd. 2010). (S5) Dataset-only ve birleşik mod çoklu test koşuları n=10 tekrar ile tamamlanmıştır (Çizelge 4.7–4.9). Wilcoxon otomatik dışa aktarımı yapılmamıştır (Wilcoxon 1945). (S6) İterasyon bazlı yakınsama eğrileri (best-so-far  $F(x)$  veya tr) raporlanmamıştır. Erken durma ve bütçe–kalite eğrisi analizi yapılmamıştır (Eiben vd. 2015).

Gelecek çalışmalar: (G1) Saha 5G/uç test yatağında  $F(x)$  kalibrasyonu. (G2) RL tabanlı aktarım ile aynı standart öznetelik şemasında hibrit karşılaştırma. (G3) Kısmi aktarım ve çoklu görev kuyruğu. (G4) EdgeCloudSim köprüsü ile topoloji duyarlı gecikme. (G5) Zaman damgalı veri seti bölme ve anlamlılık testleri. (G6) Senaryo–şartlı hiperparametre

meta-öğrenme. (G7) Yakınsama eğrileri ve erken durma ile bütçe duyarlılık analizi (Bölüm 3.3.15–3.3.16).

Meta-sezgisel algoritmaların MEC'de kullanım uygunluğu, NP-hard optimizasyon problemlerinde deterministik yöntemlerin yetersiz kalması ve gerçek zamanlı karar gereksinimleriyle açıklanır (Taleb vd. 2017). Bu algoritmalar, görev zamanlama ve yük dengeleme bağlamında yaklaşık çözümler üreterek sistemin dinamik koşullara adaptasyonuna katkı sağlarlar (Rodrigues vd. 2020). Bu tezdeki simülasyon tabanlı kanıt, aynı standart öznitelik şeması ve amaç fonksiyonu altında beş ailenin karşılaştırılmasıyla bu iddiayı somutlaştırmıştır: operatör, yoğunluk ve kaynak profiline göre GA/ABC veya PSO/GWO seçerek gecikme–kaynak dengesini iyileştirebilir (Mach vd. 2017) (Mao vd. 2017). Gelecekte RL ile hibrit politika öğrenimi, bu sezgisel seçim çerçevesinin üzerine inşa edilebilir (Zhang vd. 2019); mevcut çalışma bu birleşim için temel karşılaştırma zeminini sunmaktadır.

## 5. SONUÇ

Bu tez çalışmasında, mobil uç bilişim ortamında hesaplama aktarım kararının karar değişkeni vektörü  $x$  ile nasıl modellenebileceği ve farklı iş yükü koşullarında hangi meta-sezgisel yaklaşımın öne çıktığı araştırılmıştır. Çalışma kapsamında tek görev–tek karar ilkesi benimsenmiş; her aday çözüm için yerel ve uç yürütme yolları aynı amaç fonksiyonu altında değerlendirilmiştir.

Çalışma kapsamında öncelikle MEC hesaplama aktarımı için tekrarlanabilir bir simülatör ve Poisson tabanlı on beş senaryo üretici geliştirilmiştir. Beş gerçek veri setinden görev havuzu türeten adaptör hattı kurulmuş; Alibaba, Azure, GeoLife, 5G ölçümü ve MLPerf kayıtları simülatörün işleyebileceği Task nesnesine dönüştürülmüştür. Genetik algoritma, parçacık sürü optimizasyonu, gri kurt algoritması, yapay arı kolonisi ve karınca kolonisi optimizasyonu ile iki referans politika, ortak arama bütçesi ve ortak değerlendirme protokolü altında karşılaştırılmıştır.

Deneysel değerlendirme iki tamamlayıcı ekseninde yürütülmüştür. Birinci ekseninde on beş Poisson senaryosu ile sentetik iş yükü profilleri; ikinci ekseninde gerçek veri kaynaklı görev havuzları kullanılmıştır. Tekli koşullar keşif amaçlı yön tayini için; çoklu testler ve en az on tekrarlı koşullar ortalama ve standart sapma raporlaması için uygulanmıştır. Dataset-only, birleşik ve hibrit modlar ile istatistiksel karşılaştırma testleri aynı karar çerçevesi içinde değerlendirilmiştir.

Elde edilen sonuçlar, tüm koşullarda tek bir üstün algoritma bulunmadığını göstermiştir. Poisson senaryolarında ve gerçek veri koşullarında iş yükü profili değiştikçe en düşük gecikme değerini veren algoritmanın da değiştiği gözlenmiştir. Yüksek kaynak ve gecikme duyarlı profillerde parçacık sürü optimizasyonu ve gri kurt algoritması daha düşük ortalama gecikme değerleri üretmiş; düşük kaynaklı veya enerji kısıtlı profillerde genetik algoritma ve yapay arı kolonisi daha düşük gecikme değerleri gözlenmiştir. Karınca kolonisi optimizasyonu ise orta kaynaklı koşullarda feromon arama maliyeti nedeniyle daha yüksek gecikme değerleri üretmiştir.

İki deneysel eksen birlikte değerlendirildiğinde, aynı karar problemi altında karşılaştırılabilir sıralamalar elde edilmiştir. Mutlak süre değerleri eksenler arasında bire bir örtüşmese de profil sınıfına göre algoritma önceliği düzeyinde tutarlı eğilimler görülmüştür. Referans politika koşullarına kıyasla iyileşme hem sentetik senaryolarda hem gerçek veri koşullarında gözlenmiştir. Bu bulgular doğrultusunda iş yükü profiline göre algoritma seçimini adım adım yönlendiren bir çerçeve oluşturulmuş ve özet çizelgelerle sunulmuştur.

Çalışmanın temel çıktıları dört başlıkta toplanabilir. Birincisi, hesaplama aktarım kararını uçtan uca test edebilen simülatör ve senaryo üretim ortamının kurulmasıdır. İkincisi, beş meta-sezgisel algoritmanın aynı protokol altında sistematik karşılaştırılmasıdır. Üçüncüsü, gerçek veri setlerinin simülatör girdisine dönüştürülmesini sağlayan veri işleme hattının tanımlanmasıdır. Dördüncüsü ise profil özelliklerine göre algoritma ailesi seçimini destekleyen uygulama önerilerinin formüle edilmesidir.

Uygulama açısından önerilen süreç şu adımlardan oluşmaktadır: iş yükü profilinin tanımlanması, birincil metriğin belirlenmesi, tekli koşu ile ön değerlendirme, çoklu test ve tekrarlı koşu ile doğrulama, gerektiğinde istatistiksel test ile sonuçların sınırlandırılması. Bu akış doğrudan sabit politika önermek yerine operatörlere karar destek sunmaktadır. Erken aşamadaki üç algoritmali deneyler ile genişletilmiş beş algoritmali protokol arasında temel bulgu yönü korunmuş; değerlendirme katmanları ve veri çeşitliliği artırılmıştır.

Çalışmanın sınırları da açık biçimde belirlenmiştir. Yanıt süresi ve kaynak metrikleri simülatör formüllerinden türetilmiştir; mutlak değerler yalnızca iç tutarlılık amacıyla yorumlanmalıdır. Tek görev–tek karar modeli eşzamanlı çoklu görevleri ve kısmi aktarımı kapsamamaktadır. Gerçek veri adaptöründeki alan eşlemesi tercihleri, sonuçların bir bölümünün veri kaynağına özgü olduğunu göstermektedir.

Sonuç olarak bu tez, mobil uç bilişimde hesaplama aktarımının bağlama duyarlı bir optimizasyon problemi olduğunu ortaya koymuştur. Geliştirilen simülatör, sentetik senaryo seti ve gerçek veri tabanlı deneyler birlikte değerlendirildiğinde, meta-sezgisel

yöntemlerin uygulanabilir ancak evrensel olmayan çözümler sunduğu görülmüştür. Elde edilen bulgular, algoritma seçiminin iş yükü profili, öncelik metriği ve değerlendirme katmanına göre yapılandırılması gerektiğini göstermektedir.

## KAYNAKLAR

- Abdullaev, I., Prodanova, N., Bhaskar, K. A., Lydia, E. L., Kadry, S., & Kim, J. (2023). Task Offloading and Resource Allocation in IoT Based Mobile Edge Computing Using Deep Learning. *Computers, Materials & Continua*, 76(2), 1463–1477. doi:10.32604/cmc.2023.038417
- Alam, M. G. R., Hassan, M. M., Uddin, M. Zi., Almogren, A., & Fortino, G. (2019). Autonomic Computation Offloading in Mobile Edge for IoT Applications. *Future Generation Computer Systems*, 90, 149–157.
- Ale, L., Zhang, N., Wu, H., Chen, D., & Han, T. (2019). Online Proactive Caching in Mobile Edge Computing Using Bidirectional Deep Recurrent Neural Network. *IEEE Internet of Things Journal*, 6(3), 5520–5530.
- Alhaidari, F., Rahman, A., & Zagrouba, R. (2023). Cloud of Things: architecture, applications and challenges. *J Ambient Intell Human Comput*, 14, 5957–5975.
- Al-Shuwaili, A., & Simeone, O. (2017). Energy-Efficient Resource Allocation for Mobile Edge Computing-Based Augmented Reality Applications. *IEEE Wireless Communications Letters*, 6(3), 398–401.
- Ashton, K. (2009). That ``Internet of Things’’ Thing: In the Real World Things Matter More than Ideas. *RFID Journal*.
- Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, 79(1–2), 151–175.
- Cao, B., Zhang, L., Li, Y., Feng, D., & Cao, W. (2019). Intelligent Offloading in Multi-Access Edge Computing: A State-of-the-Art Review and Framework. *IEEE Communications Magazine*, 57(3), 56–62.
- Carvalho, G., Cabral, B., Pereira, V., & Bernardino, J. (2020). Computation offloading in Edge Computing environments using Artificial Intelligence techniques. *Engineering Applications of Artificial Intelligence*, 95, 103840.
- Chen, J., Chen, S., Wang, Q., Cao, B., Feng, G., & Hu, J. (2019). iRAF: A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks. *IEEE Internet of Things Journal*, 6(4), 7011–7024.
- Chen, J., & Ran, X. (2019). Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*, 1–20.
- Chen, Y., Gu, W., Xu, J., Zhang, Y., & Min, G. (2023). Dynamic task offloading for digital twin-empowered mobile edge computing via deep reinforcement learning. *China Communications*, 20(11), 164–175.
- Cheng, N., Lyu, F., Quan, W., Zhou, C., He, H., Shi, W., & Shen, X. (2019). Space/Aerial-Assisted Computing Offloading for IoT Applications: A Learning-Based Approach. *IEEE Journal on Selected Areas in Communications*, 37(5), 1117–1129. doi:10.1109/JSAC.2019.2906789

- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Lawrence Erlbaum Associates.
- Deng, S., Xiang, Z., Taheri, J., Mohammad, K. A., Yin, J., Zomaya, A., & Dustdar, S. (2020). Optimal Application Deployment in Resource Constrained Distributed Edges. *IEEE Transactions on Mobile Computing*.
- Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
- Dolui, K., & Datta, S. K. (2017). Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. *2017 Global Internet of Things Summit (GloTS)*, 1–6.
- Dong, R., She, C., Hardjawana, W., Li, Y., & Vucetic, B. (2019). Deep Learning for Hybrid 5G Services in Mobile Edge Computing Systems: Learn From a Digital Twin. *IEEE Transactions on Wireless Communications*, 18(10), 4692–4707.
- Dorigo, M. (1992). *Optimization, learning and natural algorithms* (Ph.D. thesis). Politecnico di Milano, Italy.
- Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *BioSystems*, 43(2), 73–81.
- Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4), 28–39.
- Eiben, A. E., & Smith, J. E. (2015). *Introduction to Evolutionary Computing* (2nd ed.). Springer.
- Farahbakhsh, F., Shahidinejad, A., & Ghobaei-Arani, M. (2023). Context-Aware computation offloading for mobile edge computing. *J Ambient Intell Human Comput*, 14, 5123–5135.
- Feng, J., Yu, F. R., Pei, Q., Chu, X., Du, J., & Zhu, L. (2020). Cooperative Computation Offloading and Resource Allocation for Blockchain-Enabled Mobile-Edge Computing: A Deep Reinforcement Learning Approach. *IEEE Internet of Things Journal*, 7(7), 6214–6228.
- Gao, H., Wang, X., Wei, W., Al-Dulaimi, A., & Xu, Y. (2024). Com-DDPG: Task Offloading Based on Multiagent Reinforcement Learning for Information-Communication-Enhanced Mobile Edge Computing in the Internet of Vehicles. *IEEE Transactions on Vehicular Technology*, 73(1), 348–361.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
- Guo, H., & Liu, J. (2018). Collaborative Computation Offloading for Multiaccess Edge Computing Over Fiber–Wireless Networks. *IEEE Transactions on Vehicular Technology*, 67(5), 4514–4526.
- Guo, J., Song, Z., Cui, Y., Liu, Z., & Ji, Y. (2017). Energy-Efficient Resource

- Allocation for Multi-User Mobile Edge Computing. *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*.
- Guo, J., Chang, Z., Wang, S., Ding, H., Feng, Y., Mao, L., & Bao, Y. (2019). Who Limits the Resource Efficiency of My Datacenter: An Analysis of Alibaba Datacenter Traces. *IEEE/ACM International Symposium on Quality of Service (IWQoS)*. doi:10.1145/3326285.3329074
- Gupta, H., Vahid Dastjerdi, A., Ghosh, S. K., & Buyya, R. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in Internet of Things, Edge and Fog computing environments. *Software: Practice and Experience*, 47(9), 1275–1296.
- Hadary, O., Marshall, L., Menache, I., Pan, A., Greeff, E. E., Dion, D., Dorminey, S., Joshi, S., Chen, Y., Russinovich, M., & Moscibroda, T. (2020). Protean: VM Allocation Service at Scale. *Proc. 14th USENIX Symp. Operating Systems Design and Implementation (OSDI)*, 845–861.
- Hamdan, S., Ayyash, M., & Almajali, S. (2020). Edge-Computing Architectures for Internet of Things Applications: A Survey. *Sensors*, 20(18), 6441.
- Hao, P., Hu, L., Jiang, J., Hu, J., & Che, X.-L. (2018). Mobile Edge Provision with Flexible Deployment. *IEEE Transactions on Services Computing*.
- Hao, Y., Chen, M., Hu, L., Hossain, M. S., & Ghoneim, A. (2018). Energy Efficient Task Caching and Offloading for Mobile Edge Computing. *IEEE Access*, 6, 11365–11373.
- He, Z., Li, K., Li, K., & Zhou, W. (2021). Server configuration optimization in mobile edge computing: A cost-performance tradeoff perspective. *Software: Practice and Experience*, 51(9), 1868–1895.
- Hermann, K. M., et al. (2015). Teaching Machines to Read and Comprehend. *Advances in Neural Information Processing Systems*.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.
- Hosseinzadeh, M., Masdari, M., Rahmani, A. M., Mohammadi, M., Aldalwie, A. H. M., Majeed, M. K., & Karim, S. H. T. (2021). Improved Butterfly Optimization Algorithm for Data Placement and Scheduling in Edge Computing Environments. *Journal of Grid Computing*, 19(2), 1463–1477.
- Huang, L., Feng, X., Zhang, C., Qian, L., & Wu, Y. (2018). Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing. *Digital Communications and Networks*.
- Huang, Y.-Y., & Wang, P.-C. (2023). Computation Offloading and User-Clustering Game in Multi-Channel Cellular Networks for Mobile Edge Computing. *Sensors*, 23(3), 1155.
- Ishaq, M., Afzal, M. H., Tahir, S., & Ullah, K. (2021). A Compact Study of Recent Trends of Challenges and Opportunities in Integrating Internet of Things (IoT) and Cloud Computing. *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 1–4.

- Jang, J., Tulkinbekov, K., & Kim, D.-H. (2023). Task Offloading of Deep Learning Services for Autonomous Driving in Mobile Edge Computing. *Electronics*, 12(3223).
- Karaboğa, D. (2005). Sayısal optimizasyon için arı kolonisi temelli yeni bir yaklaşıma ait fikir. Teknik Rapor-TR06, Erciyes Üniversitesi, Türkiye.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459-471.
- Kasi, M. K., Abu Ghazalah, S., Akram, R. N., & Sauveron, D. (2021). Secure Mobile Edge Server Placement Using Multi-Agent Reinforcement Learning. *Electronics*, 10(17), 2098.
- Kavitha, B. C., Vallikannu, A. L., & Sankaran, K. S. (2020). Delay-Aware Concurrent Data Management Method For IoT Collaborative Mobile Edge Computing Environment. *Microprocessors and Microsystems*, 103021.
- Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1942–1948.
- Lähderanta, T., Leppänen, T., Ruha, L., Lovén, L., Harjula, E., Ylianttila, M., & Sillanpää, M. J. (2021). Edge computing server placement with capacitated location allocation. *Journal of Parallel and Distributed Computing*, 153, 130–149.
- Li, B., Hou, P., Wang, K., Peng, Z., Jin, S., & Niu, L. (2020). Deployment of edge servers in 5G cellular networks. *Transactions on Emerging Telecommunications Technologies*.
- Li, B., Hou, P., Wu, H., & Hou, F. (2021). Optimal edge server deployment and allocation strategy in 5G ultra-dense networking environments. *Pervasive and Mobile Computing*, 72, 101312.
- Liao, H., Zhou, Z., Zhao, X., Zhang, L., Mumtaz, S., Jolfaei, A., & Bashir, A. K. (2019). Learning-Based Context-Aware Resource Allocation for Edge Computing-Empowered Industrial IoT. *IEEE Internet of Things Journal*, 1.
- Liao, L., Lai, Y., Yang, F., & Zeng, W. (2023). Online computation offloading with double reinforcement learning algorithm in mobile edge computing. *Journal of Parallel and Distributed Computing*, 171, 28–39.
- Liu, C.-F., Bennis, M., & Poor, H. v. (2017). Latency and Reliability-Aware Task Offloading and Resource Allocation for Mobile Edge Computing. 2017 IEEE Globecom Workshops (GC Wkshps).
- Liu, Y., Peng, M., Shou, G., Chen, Y., & Chen, S. (2020). Toward Edge Intelligence: Multiaccess Edge Computing for 5G and Internet of Things. *IEEE Internet of Things Journal*, 7(8), 6722–6747.
- Liu, Y., & Peng, C. (2023). A Close Look at 5G in the Wild: Unrealized Potentials and Implications. *IEEE International Conference on Computer Communications (INFOCOM)*.
- Liu, Z., Zhang, J., Li, Y., & Ji, Y. (2020). Hierarchical MEC Servers Deployment

- and User-MEC Server Association in C-RANs over WDM Ring Networks. *Sensors*, 20(5), 1282.
- Mach, P., & Becvar, Z. (2017). Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys & Tutorials*, 19(3), 1628–1656.
- Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials*, 19(4), 2322–2358.
- Mao, Y., Zhang, J., Song, S., & Letaief, K. B. (2017). Stochastic Joint Radio and Computational Resource Management for Multi-User Mobile-Edge Computing Systems. *IEEE Transactions on Wireless Communications*, 16(3), 1488–1503.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46–61.
- Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *IEEE Communications Surveys & Tutorials*, 20(4), 2923–2960.
- Mora-Gimeno, F. J., Mora-Mora, H., Marcos-Jorquera, D., & Volckaert, B. (2018). A Secure Multi-Tier Mobile Edge Computing Model for Data Processing Offloading Based on Degree of Trust. *Sensors*, 18(10), 3211.
- Pham, Q.-V., le Anh, T., Tran, N. H., Park, B. J., & Hong, C. S. (2018). Decentralized Computation Offloading and Resource Allocation for Mobile-Edge Computing: A Matching Game Approach. *IEEE Access*.
- Rao, S. S. (2009). *Engineering Optimization: Theory and Practice* (4th ed.). John Wiley & Sons.
- Rath, M., Satpathy, J., & Oreku, G. S. (2021). Chapter 6 - Artificial Intelligence and Machine Learning Applications in Cloud Computing and Internet of Things. In *Artificial Intelligence to Solve Pervasive Internet of Things Issues* (pp. 103–123). Academic Press.
- Reddi, V. J., Cheng, C., Kanter, D., Mattson, P., et al. (2020). MLPerf Inference Benchmark. *Proc. ACM/IEEE 47th Annu. Int. Symp. Computer Architecture (ISCA)*, 446–459. (arXiv:1911.02549).
- Rimal, B. P., Pham Van, D., & Maier, M. (2017). Cloudlet Enhanced Fiber-Wireless Access Networks for Mobile-Edge Computing. *IEEE Transactions on Wireless Communications*, 16(6), 3601–3618.
- Rodrigues, T. K., Suto, K., Nishiyama, H., Liu, J., & Kato, N. (2020). Machine Learning Meets Computation and Communication Control in Evolving Edge and Cloud: Challenges and Future Perspective. *IEEE Communications Surveys & Tutorials*, 22(1), 38–67.
- Shakarami, A., Ghobaei-Arani, M., & Shahidinejad, A. (2020). A Survey on the Computation Offloading Approaches in Mobile Edge Computing: A Machine Learning-based Perspective. *Computer Networks*, 182, 107496.
- Shao, M., Liu, J., Yang, Q., & Simon, G. (2020). A Learning Based Framework for

- MEC Server Planning With Uncertain BSs Demands. *IEEE Access*, 8, 198832–198844.
- Shi, Y., & Liu, J. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5), 637–646.
- Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation Proceedings, 69–73.
- Singh, N., Raza, M., Paranthaman, V. V., Awais, M., Khalid, M., & Javed, E. (2021). Chapter 10 - Internet of Things and cloud computing. In *Digital Health* (pp. 151–162). Academic Press.
- Sohaib, M., Jeon, S.-W., & Yu, W. (2024). Hybrid Online-Offline Learning for Task Offloading in Mobile Edge Computing Systems. arXiv:2402.11743.
- Sonmez, C., Ozgovde, A., & Ersoy, C. (2017). EdgeCloudSim: An Environment for Performance Evaluation of Edge Computing Systems. Proc. 2nd Int. Conf. Fog and Mobile Edge Computing (FMEC), 39–44.
- Sun, W., Liu, J., Yue, Y., & Zhang, H. (2018). Double Auction-based Resource Allocation for Mobile Edge Computing in Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*.
- Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, 19(3), 1657–1681.
- Tran, T. X., & Pompili, D. (2018). Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks. *IEEE Transactions on Vehicular Technology*.
- Vimal, S., Khari, M., Dey, N., Crespo, R. G., & Harold Robinson, Y. (2020). Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT. *Computer Communications*, 151, 355–364.
- Wang, C., Liang, C., Yu, F. R., Chen, Q., & Tang, L. (2017). Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing. *IEEE Transactions on Wireless Communications*, 16(8), 4924–4938.
- Wang, E., Li, D., Dong, B., Zhou, H., & Zhu, M. (2020). Flat and hierarchical system deployment for edge computing systems. *Future Generation Computer Systems*, 105, 308–317.
- Wang, J., Zhao, L., Liu, J., & Kato, N. (2019a). Smart Resource Allocation for Mobile Edge Computing: A Deep Reinforcement Learning Approach. *IEEE Transactions on Emerging Topics in Computing*. doi:10.1109/tetc.2019.2902661
- Wang, S., Zhao, Y., Huang, L., Xu, J., & Hsu, C.-H. (2019b). QoS prediction for service recommendations in mobile edge computing. *Journal of Parallel and Distributed Computing*. doi:10.1016/j.jpdc.2017.09.014

- Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., & Chen, M. (2019c). In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning. *IEEE Network*. doi:10.1109/MNET.2019.1800286
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80–83.
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82.
- Wu, W., Zhou, F., Hu, R. Q., & Wang, B. (2019). Energy-Efficient Resource Allocation for Secure NOMA-Enabled Mobile Edge Computing Networks. *IEEE Transactions on Communications*.
- Xiang, Z., Deng, S., Taheri, J., & Zomaya, A. (2019). Dynamical Service Deployment and Replacement in Resource-Constrained Edges. *Mobile Networks and Applications*.
- Xiao, L., Wan, X., Dai, C., Du, X., Chen, X., & Guizani, M. (2018). Security in Mobile Edge Caching with Reinforcement Learning. *IEEE Wireless Communications*, 25(3), 116–122.
- Yang, L., Zhang, H., Li, X., Ji, H., & Leung, V. C. M. (2018). A Distributed Computation Offloading Strategy in Small-Cell Networks Integrated With Mobile Edge Computing. *IEEE/ACM Transactions on Networking*.
- You, C., Huang, K., Chae, H., & Kim, B.-H. (2017). Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading. *IEEE Transactions on Wireless Communications*, 16(3), 1397–1411.
- Zaman, S. K. u., Jehangiri, A. I., Maqsood, T., & et al. (2023). LiMPO: lightweight mobility prediction and offloading framework using machine learning for mobile edge computing. *Cluster Comput*, 26, 99–117.
- Zhang, C., & Zheng, Z. (2019). Task migration for mobile edge computing using deep reinforcement learning. *Future Generation Computer Systems*, 96, 111–118. doi:10.1016/j.future.2019.01.059
- Zhang, J., Xia, W., Yan, F., & Shen, L. (2018). Joint Computation Offloading and Resource Allocation Optimization in Heterogeneous Networks With Mobile Edge Computing. *IEEE Access*, 6, 19324–19337.
- Zhang, K., Zhu, Y., Leng, S., He, Y., Maharjan, S., & Zhang, Y. (2019). Deep Learning Empowered Task Offloading for Mobile Edge Computing in Urban Informatics. *IEEE Internet of Things Journal*, 6(5), 7635–7647.
- Zhao, R., Wang, X., Xia, J., & Fan, L. (2020). Deep reinforcement learning based mobile edge computing for intelligent Internet of Things. *Physical Communication*.
- Zhao, T., Zhou, S., Guo, X., & Niu, Z. (2017). Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing. *2017 IEEE International Conference on Communications (ICC)*.
- Zheng, Y., Xie, X., & Ma, W.-Y. (2010). GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data*

Engineering Bulletin, 33(2), 32–39.

Zhou, F., Hu, R. Q., Li, Z., & Wang, Y. (2020). Mobile Edge Computing in Unmanned Aerial Vehicle Networks. *IEEE Wireless Communications*, 27(1), 140–146.

Zhou, J., Fan, J., Wang, J., & Jia, J. (2019). Dynamic service deployment for budget-constrained mobile edge computing. *Concurrency and Computation: Practice and Experience*.

## **EKLER**

**EK 1 Gerçek Dünya Veri Setleri, Atıflar ve Deney Protokolü**

**EK 2 On Beş Senaryo ScenarioConfig Parametreleri**

**EK 3 Gerçek Dünya Veri Setleri Değişken Uyumluluğu**

## EK 1 Gerçek Dünya Veri Setleri, Atıflar ve Deney Protokolü

Bu ekte, Bölüm 3.5.3 ve 4.1.3'te özetlenen gerçek dünya veri setleri deney hattının atıf bilgileri, ham kaynak özeti ve ortak deney protokolü tablo halinde sunulmaktadır; tüm atıflar birincil kaynaklara dayanmaktadır. Tez kapsamında kullanılan uygulama, veri ve sonuçlara <https://github.com/speedygonzalo/MecClient> ve <https://github.com/speedygonzalo/MecServer> repolarından ulaşılabilmektedir.

Gerçek dünya veri setleri, atıflar ve deney protokolü

| Veri seti                   | Kod etiketi  | Birincil atıf                       | Ham kaynak özeti                         | Deney protokolü                                                           |
|-----------------------------|--------------|-------------------------------------|------------------------------------------|---------------------------------------------------------------------------|
| Alibaba Cluster Trace v2018 | ALIBABA_V2   | Guo vd. (2019)                      | batch_task, machine_meta, container_meta | require_real_data=True; drop_missing_rows; clamp; F(x); pop=100; iter=100 |
| Azure VM Packing            | AZURE_PUBLIC | Hadary vd. (2020)                   | AzurePackingTraceV1 SQLite               | Aynı protokol; dataset-only mod                                           |
| CNN/DailyMail (MLPerf Edge) | MLPERF_EDGE  | Herman vd. (2015); Reddi vd. (2020) | Hugging Face cnn_dailymail               | Çıkarım yükü vektörü; adaptör kalibrasyonu                                |
| INFOCOM'23 5G ölçümü        | 5G_MONARCH   | Liu ve Peng (2023)                  | Grid lat/lon, kanal güçleri CSV          | Ağ koşulları ağırlıklı profil                                             |
| Microsoft GeoLife GPS       | CRAWDAD      | Zheng vd. (2010)                    | GeoLife .plt trajectory                  | Mobilite/ coğrafi türetim                                                 |

## EK 2 On Beş Senaryo ScenarioConfig Parametreleri

Poisson tabanlı veri üreticinin kullandığı on beş senaryonun merkez değerleri ( $\lambda$ ) config.py dosyasındaki SCENARIO\_CONFIGS sözlüğünden alınmış olup, değişken gruplarına göre altı çizelgede sunulmaktadır (Bölüm 3.1.1).

Coğrafi değişken grubuna ait ScenarioConfig merkez değerleri Çizelge 'de görselleştirilmiştir.

ScenarioConfig Coğrafi merkez değerleri (on beş senaryo)

| Senaryo                                        | distance | terrain_type | terrain_harshness | weather | user_density |
|------------------------------------------------|----------|--------------|-------------------|---------|--------------|
| TEMEL (BASIC)                                  | 100      | 5            | 5                 | 5       | 50           |
| HIGH_DENSITY_URBAN (Yüksek Yoğunluklu Kentsel) | 50       | 8            | 7                 | 6       | 500          |
| SMART_CITY (Akıllı Şehir)                      | 75       | 7            | 6                 | 6       | 300          |
| BUSINESS_DISTRICT (İş Bölgesi)                 | 60       | 6            | 5                 | 5       | 200          |
| EVENT_DRIVEN (Etkinlik Kaynaklı)               | 40       | 7            | 6                 | 6       | 1000         |
| LIMITED_INTERNET (Sınırlı İnternet)            | 150      | 9            | 8                 | 7       | 20           |
| AGRICULTURAL (Tarımsal Ağ)                     | 120      | 8            | 7                 | 6       | 30           |
| FIBER_5G (Fiber/5G Kırsal)                     | 100      | 6            | 5                 | 5       | 50           |
| TELEMEDICINE (Teletıp)                         | 80       | 5            | 4                 | 4       | 40           |
| SUBURBAN_MIXED (Banliyö Karma)                 | 90       | 6            | 5                 | 5       | 150          |
| WORK_FROM_HOME (Evden Çalışma)                 | 85       | 5            | 4                 | 4       | 120          |
| DATA_CENTER (Veri Merkezi)                     | 30       | 4            | 3                 | 3       | 2000         |
| DISASTER_RECOVERY (Felaket Kurtarma)           | 100      | 8            | 7                 | 8       | 500          |
| EDGE_COMPUTING (Uzak Bölge Uç)                 | 70       | 6            | 5                 | 5       | 80           |
| TEMPORARY_NETWORK (Geçici Ağ)                  | 90       | 7            | 6                 | 6       | 100          |

Görev değişken grubuna ait ScenarioConfig merkez değerleri Çizelge'de görselleştirilmiştir.

ScenarioConfig Görev merkez değerleri (on beş senaryo)

| Senaryo                                        | task type   | task size | task complexity | recurrence | delay tolerance |
|------------------------------------------------|-------------|-----------|-----------------|------------|-----------------|
| TEMEL (BASIC)                                  | COMPUTATION | 500       | 3               | 3          | 50              |
| HIGH DENSITY URBAN (Yüksek Yoğunluklu Kentsel) | COMPUTATION | 1000      | 5               | 4          | 30              |
| SMART CITY (Akıllı Şehir)                      | MIXED       | 800       | 4               | 5          | 40              |
| BUSINESS DISTRICT (İş Bölgesi)                 | COMPUTATION | 600       | 3               | 3          | 45              |
| EVENT DRIVEN (Etkinlik Kaynaklı)               | MIXED       | 400       | 4               | 2          | 35              |
| LIMITED INTERNET (Sınırlı İnternet)            | COMPUTATION | 100       | 2               | 2          | 100             |
| AGRICULTURAL (Tarımsal Ağ)                     | MIXED       | 150       | 3               | 3          | 80              |
| FIBER 5G (Fiber/5G Kırsal)                     | COMPUTATION | 900       | 4               | 4          | 40              |
| TELEMEDICINE (Teletıp)                         | MIXED       | 800       | 5               | 3          | 30              |
| SUBURBAN MIXED (Banliyö Karma)                 | MIXED       | 700       | 3               | 3          | 45              |
| WORK FROM HOME (Evden Çalışma)                 | COMPUTATION | 600       | 4               | 4          | 40              |
| DATA CENTER (Veri Merkezi)                     | COMPUTATION | 1200      | 6               | 5          | 20              |
| DISASTER RECOVERY (Felaket Kurtarma)           | MIXED       | 500       | 5               | 2          | 60              |
| EDGE COMPUTING (Uzak Bölge Uç)                 | COMPUTATION | 400       | 4               | 3          | 35              |
| TEMPORARY_NETWORK (Geçici Ağ)                  | MIXED       | 300       | 3               | 2          | 45              |

Yerel kaynak deęiřken grubuna ait ScenarioConfig merkez deęerleri izelge grselleřtirilmiřtir.

ScenarioConfig Yerel kaynak merkez deęerleri (on beř senaryo)

| Senaryo                                           | cpu_power | memory | hdd_space | network_adapter_speed | bus_speed |
|---------------------------------------------------|-----------|--------|-----------|-----------------------|-----------|
| TEMEL (BASIC)                                     | 70        | 70     | 70        | 70                    | 70        |
| HIGH_DENSITY_URBAN<br>(Yksek Yoęunluklu Kentsel) | 80        | 80     | 80        | 80                    | 80        |
| SMART_CITY (Akıllı řehir)                         | 75        | 75     | 75        | 75                    | 75        |
| BUSINESS_DISTRICT (İř Blgesi)                    | 65        | 65     | 65        | 65                    | 65        |
| EVENT_DRIVEN (Etkinlik<br>Kaynaklı)               | 60        | 60     | 60        | 60                    | 60        |
| LIMITED_INTERNET (Sınırlı<br>İnternet)            | 40        | 40     | 40        | 40                    | 40        |
| AGRICULTURAL (Tarımsal Aę)                        | 50        | 50     | 50        | 50                    | 50        |
| FIBER_5G (Fiber/5G Kırsal)                        | 85        | 85     | 85        | 85                    | 85        |
| TELEMEDICINE (Teletıp)                            | 90        | 90     | 90        | 90                    | 90        |
| SUBURBAN_MIXED (Banliy<br>Karma)                 | 70        | 70     | 70        | 70                    | 70        |
| WORK_FROM_HOME (Evden<br>alıřma)                 | 75        | 75     | 75        | 75                    | 75        |
| DATA_CENTER (Veri Merkezi)                        | 95        | 95     | 95        | 95                    | 95        |
| DISASTER_RECOVERY (Felaket<br>Kurtarma)           | 60        | 60     | 60        | 60                    | 60        |
| EDGE_COMPUTING (Uzak Blge<br>U)                 | 65        | 65     | 65        | 65                    | 65        |
| TEMPORARY_NETWORK<br>(Geici Aę)                  | 55        | 55     | 55        | 55                    | 55        |

Uç sunucu değişken grubuna ait ScenarioConfig merkez değerleri Çizelge 'de görselleştirilmiştir.

ScenarioConfig Uç sunucu merkez değerleri (on beş senaryo)

| Senaryo                                        | server_type | cpu_power_availability | memory_availability | hdd_space_availability | bus_speed_availability | network_adapter_speed_availability | network_queue | mobility | speed |
|------------------------------------------------|-------------|------------------------|---------------------|------------------------|------------------------|------------------------------------|---------------|----------|-------|
| TEMEL (BASIC)                                  | EDGE        | 70                     | 70                  | 70                     | 70                     | 70                                 | 10            | 2        | 2     |
| HIGH_DENSITY_URBAN (Yüksek Yoğunluklu Kentsel) | EDGE        | 80                     | 80                  | 80                     | 80                     | 80                                 | 15            | 3        | 3     |
| SMART_CITY (Akıllı Şehir)                      | EDGE        | 75                     | 75                  | 75                     | 75                     | 75                                 | 12            | 2        | 2     |
| BUSINESS_DISTRICT (İş Bölgesi)                 | LOCAL       | 65                     | 65                  | 65                     | 65                     | 65                                 | 8             | 1        | 1     |
| EVENT_DRIVEN (Etkinlik Kaynaklı)               | EDGE        | 60                     | 60                  | 60                     | 60                     | 60                                 | 25            | 4        | 4     |
| LIMITED_INTERNET (Sınırlı İnternet)            | LOCAL       | 40                     | 40                  | 40                     | 40                     | 40                                 | 5             | 1        | 1     |
| AGRICULTURAL (Tarımsal Ağ)                     | LOCAL       | 50                     | 50                  | 50                     | 50                     | 50                                 | 6             | 1        | 1     |
| FIBER_5G (Fiber/5G Kırsal)                     | EDGE        | 85                     | 85                  | 85                     | 85                     | 85                                 | 8             | 2        | 2     |
| TELEMEDICINE (Teletıp)                         | EDGE        | 90                     | 90                  | 90                     | 90                     | 90                                 | 5             | 1        | 1     |
| SUBURBAN_MIXED (Banliyö Karma)                 | EDGE        | 70                     | 70                  | 70                     | 70                     | 70                                 | 10            | 2        | 2     |
| WORK_FROM_HOME (Evden Çalışma)                 | LOCAL       | 75                     | 75                  | 75                     | 75                     | 75                                 | 8             | 1        | 1     |
| DATA_CENTER (Veri Merkezi)                     | EDGE        | 95                     | 95                  | 95                     | 95                     | 95                                 | 3             | 1        | 1     |
| DISASTER_RECOVERY (Felaket Kurtarma)           | EDGE        | 60                     | 60                  | 60                     | 60                     | 60                                 | 15            | 3        | 3     |
| EDGE_COMPUTING (Uzak Bölge Uç)                 | EDGE        | 65                     | 65                  | 65                     | 65                     | 65                                 | 7             | 2        | 2     |
| TEMPORARY_NETWORK (Geçici Ağ)                  | EDGE        | 55                     | 55                  | 55                     | 55                     | 55                                 | 6             | 3        | 3     |

Ağ değişken grubuna ait ScenarioConfig merkez değerleri Çizelge 'de görselleştirilmiştir.

ScenarioConfig Ağ merkez değerleri (on beş senaryo)

| Senaryo                                        | connection_type | connection_speed | connection_reliability | node_count |
|------------------------------------------------|-----------------|------------------|------------------------|------------|
| TEMEL (BASIC)                                  | 2               | 500              | 80                     | 3          |
| HIGH_DENSITY_URBAN (Yüksek Yoğunluklu Kentsel) | 5               | 1000             | 95                     | 10         |
| SMART_CITY (Akıllı Şehir)                      | 4               | 800              | 90                     | 8          |
| BUSINESS_DISTRICT (İş Bölgesi)                 | 3               | 600              | 85                     | 5          |
| EVENT_DRIVEN (Etkinlik Kaynaklı)               | 3               | 400              | 70                     | 15         |
| LIMITED_INTERNET (Sınırlı İnternet)            | 1               | 100              | 50                     | 2          |
| AGRICULTURAL (Tarımsal Ağ)                     | 2               | 150              | 60                     | 3          |
| FIBER_5G (Fiber/5G Kırsal)                     | 5               | 900              | 95                     | 5          |
| TELEMEDICINE (Teletıp)                         | 5               | 800              | 90                     | 4          |
| SUBURBAN_MIXED (Banliyö Karma)                 | 4               | 700              | 85                     | 6          |
| WORK_FROM_HOME (Evden Çalışma)                 | 4               | 600              | 80                     | 4          |
| DATA_CENTER (Veri Merkezi)                     | 5               | 1200             | 98                     | 35         |
| DISASTER_RECOVERY (Felaket Kurtarma)           | 3               | 500              | 45                     | 10         |
| EDGE_COMPUTING (Uzak Bölge Uç)                 | 4               | 400              | 75                     | 5          |
| TEMPORARY_NETWORK (Geçici Ağ)                  | 3               | 300              | 70                     | 4          |

Dış çevre değişken grubuna ait ScenarioConfig merkez değerleri Çizelge 'de görselleştirilmiştir.

ScenarioConfig Dış çevre merkez değerleri (on beş senaryo)

| Senaryo                                        | user_speed | user_direction | crowd_frequency | crowd_usage |
|------------------------------------------------|------------|----------------|-----------------|-------------|
| TEMEL (BASIC)                                  | 5          | 180            | 30              | 30          |
| HIGH_DENSITY_URBAN (Yüksek Yoğunluklu Kentsel) | 8          | 180            | 90              | 85          |
| SMART_CITY (Akıllı Şehir)                      | 6          | 180            | 80              | 75          |
| BUSINESS_DISTRICT (İş Bölgesi)                 | 4          | 180            | 60              | 70          |
| EVENT_DRIVEN (Etkinlik Kaynaklı)               | 7          | 180            | 95              | 90          |
| LIMITED_INTERNET (Sınırlı İnternet)            | 2          | 180            | 30              | 40          |
| AGRICULTURAL (Tarımsal Ağ)                     | 3          | 180            | 40              | 50          |
| FIBER_5G (Fiber/5G Kırsal)                     | 4          | 180            | 50              | 80          |
| TELEMEDICINE (Teletıp)                         | 2          | 180            | 60              | 85          |
| SUBURBAN_MIXED (Banliyö Karma)                 | 5          | 180            | 70              | 75          |
| WORK_FROM_HOME (Evden Çalışma)                 | 3          | 180            | 80              | 85          |
| DATA_CENTER (Veri Merkezi)                     | 2          | 180            | 90              | 95          |
| DISASTER_RECOVERY (Felaket Kurtarma)           | 5          | 180            | 70              | 80          |
| EDGE_COMPUTING (Uzak Bölge Uç)                 | 4          | 180            | 50              | 60          |
| TEMPORARY_NETWORK (Geçici Ağ)                  | 5          | 180            | 60              | 70          |

### EK 3 Gerçek Dünya Veri Setleri Değişken Uyumluluğu

Bu ekte, simülatörde kullanılan otuz iki MEC değişkeninin beş gerçek dünya veri setleriyle eşlemesi satır düzeyinde verilmektedir. Uyumluluk sınıfları: Doğrudan, Türetilbilir ve Yok (Bölüm 3.5.4 ile tutarlı).

Alibaba Cluster Trace v2018 (ALIBABA\_V2)

Kaynak: Guo vd. (2019). \*\*Ham alanlar:\*\* `task\_name`, `instance\_num`, `job\_name`, `task\_type`, `status`, `start\_time`, `end\_time`, `plan\_cpu`, `plan\_mem`.

Alibaba Cluster Trace v2018 değişken uyumluluğu

| Değişken               | Kaynak alan             | Uyumluluk                  | Not                                                                       |
|------------------------|-------------------------|----------------------------|---------------------------------------------------------------------------|
| Coğrafi                |                         |                            |                                                                           |
| distance               | start_time,<br>end_time | Türetilbilir               | Görev süresi → $\log_{1p}(\text{duration}) \rightarrow \text{distance}$ . |
| terrain_type           | -                       | Türetilbilir               | CPU planı coğrafi vektöre projekte edilir.                                |
| terrain harshness      | -                       | Türetilbilir               | Bellek planı.                                                             |
| weather                | -                       | Türetilbilir               | Örnek sayısı.                                                             |
| user_density           | job_name                | Türetilbilir               | İş adı uzunluğu → yoğunluk vektörü.                                       |
| Görev                  |                         |                            |                                                                           |
| task_type              | task_type<br>(ham)      | Türetilbilir               | Ham izde var; adaptör 'COMPUTATION' sabitler.                             |
| task_size              | plan_cpu,<br>plan_mem   | Doğrudan /<br>Türetilbilir | $\text{plan\_cpu} \times 8 + \text{plan\_mem} \times 12$ .                |
| task_complexity        | instance_num            | Türetilbilir               | $\log_{1p}(\text{instance}) \times 3$ .                                   |
| recurrence             | instance_num            | Türetilbilir               | Tekrar örnek sayısı ile.                                                  |
| delay_tolerance        | start_time,<br>end_time | Türetilbilir               | Görev süresi.                                                             |
| Yerel                  |                         |                            |                                                                           |
| cpu_power              | plan_cpu                | Türetilbilir               | Yerel CPU gücü vektörü.                                                   |
| memory                 | plan_mem                | Türetilbilir               | Yerel bellek.                                                             |
| hdd_space              | —                       | Yok                        | Kod sabiti 70.                                                            |
| network_adapter_speed  | plan_mem                | Türetilbilir               | Bellek ile ölçeklenir.                                                    |
| bus_speed              | —                       | Yok                        | Kod sabiti 70.                                                            |
| Sunucu                 |                         |                            |                                                                           |
| server_type            | —                       | Türetilbilir               | 'server_edge_indicator' rastgele (~%55 EDGE).                             |
| cpu_power_availability | plan_cpu                | Türetilbilir               | $100 - \text{plan\_cpu}$ .                                                |
| memory_availability    | plan_mem                | Türetilbilir               | $100 - \text{plan\_mem}$ .                                                |
| hdd_space_availability | —                       | Yok                        | Kod sabiti 70.                                                            |

|                                    |              |               |                           |
|------------------------------------|--------------|---------------|---------------------------|
| bus_speed_availability             | —            | Yok           | Kod sabiti 70.            |
| network_adapter_speed_availability | plan_mem     | Türetilebilir |                           |
| network_queue                      | instance_num | Türetilebilir | Kuyruk yoğunluğu vektörü. |
| mobility                           | —            | Yok           | Kod sabiti 1.0.           |
| speed                              | —            | Yok           | Kod sabiti 1.0.           |
| Ağ                                 |              |               |                           |
| connection_type                    | —            | Yok           | Kod sabiti 2.5.           |
| connection_speed                   | plan_cpu     | Türetilebilir | plan_cpu×20.              |
| connection_reliability             | —            | Yok           | Kod sabiti 85.            |
| node_count                         | instance_num | Türetilebilir |                           |
| Dış Ortam                          |              |               |                           |
| user_speed                         | —            | Yok           | Kod sabiti 2.0.           |
| user_direction                     | —            | Yok           | Kod sabiti 180.           |
| crowd_frequency                    | —            | Yok           | Kod sabiti 30.            |
| crowd_usage                        | —            | Yok           | Kod sabiti 30.            |

Özet: Doğrudan: 1–2, Türetilebilir: ~18, Yok (sabit/BASIC): ~12. **\*\*Güçlü boyut:\*\*** görev boyutu, kaynak planı, süre.

Azure VM Packing (AZURE\_PUBLIC)

Kaynak: Hadary vd. (2020). **\*\*Ham alanlar:\*\*** VM packing SQLite tablosu (sayısal sütunlar; CPU, bellek, disk kullanım oranları).

#### Azure VM Packing değişken uyumluluğu

| Değişken          | Kaynak alan            | Uyumluluk     | Not                            |
|-------------------|------------------------|---------------|--------------------------------|
| Coğrafi           |                        |               |                                |
| distance          | CPU + bellek vektörü   | Türetilebilir | İlk iki sayısal sütun toplamı. |
| terrain_type      | -                      | Türetilebilir | VM çekirdek yükü.              |
| terrain_harshness | -                      | Türetilebilir | Bellek baskısı.                |
| weather           | —                      | Yok           | Kod sabiti 5.0.                |
| user_density      | satır kimliği uzunluğu | Türetilebilir | İlk sütun string uzunluğu.     |
| Görev             |                        |               |                                |
| task_type         | —                      | Yok           | Adaptör 'COMPUTATION'.         |
| task_size         | core, mem              | Türetilebilir | (core+mem)×500.                |
| task_complexity   | core                   | Türetilebilir | core×10.                       |
| recurrence        | —                      | Yok           | Kod sabiti 3.0.                |
| delay_tolerance   | mem                    | Türetilebilir | mem×80.                        |
| Yerel             |                        |               |                                |
| cpu_power         | core                   | Türetilebilir | core×100.                      |

|                                    |                        |               |                                                          |
|------------------------------------|------------------------|---------------|----------------------------------------------------------|
| memory                             | mem                    | Türetilebilir | $\text{mem} \times 100$ .                                |
| hdd_space                          | 3. sayısal sütun (hdd) | Türetilebilir | Varsa disk; yoksa $\text{core} \times 0.5$ .             |
| network_adapter_speed              | —                      | Yok           | Kod sabiti 75.                                           |
| bus_speed                          | —                      | Yok           | Kod sabiti 75.                                           |
| Sunucu                             |                        |               |                                                          |
| server_type                        | —                      | Türetilebilir | `server_edge_indicator` sabit 0.55.                      |
| cpu_power_availability             | core                   | Türetilebilir | $(1 - \text{core}) \times 100$ — kenar kapasite baskısı. |
| memory_availability                | mem                    | Türetilebilir | $(1 - \text{mem}) \times 100$ .                          |
| hdd_space_availability             | —                      | Yok           | Kod sabiti 70.                                           |
| bus_speed_availability             | —                      | Yok           | Kod sabiti 70.                                           |
| network_adapter_speed_availability | —                      | Yok           | Kod sabiti 75.                                           |
| network_queue                      | core                   | Türetilebilir | $\text{core} \times 20$ .                                |
| mobility                           | —                      | Yok           | Kod sabiti 0.5.                                          |
| speed                              | —                      | Yok           | Kod sabiti 0.5.                                          |
| Ağ                                 |                        |               |                                                          |
| connection_type                    | —                      | Yok           | Kod sabiti 2.5.                                          |
| connection_speed                   | mem                    | Türetilebilir | $\text{mem} \times 5000$ .                               |
| connection_reliability             | —                      | Yok           | Kod sabiti 88.                                           |
| node_count                         | —                      | Yok           | Kod sabiti 4.0.                                          |
| Dış Ortam                          |                        |               |                                                          |
| user_speed                         | —                      | Yok           | Kod sabiti 1.0.                                          |
| user_direction                     | —                      | Yok           | Kod sabiti 90.                                           |
| crowd_frequency                    | —                      | Yok           | Kod sabiti 25.                                           |
| crowd_usage                        | —                      | Yok           | Kod sabiti 25.                                           |

Özet: Doğrudan: 0, Türetilebilir: ~16, Yok (sabit): ~16. **\*\*Güçlü boyut:\*\*** VM yerleştirme, kapasite/kuyruk baskısı.

GeoLife GPS (CRAWDAD)

Kaynak: Zheng vd. (2010). **\*\*Ham alanlar:\*\*** `.plt` trajectory — enlem, boylam, zaman damgası; ardışık noktalardan hız türetilir.

#### GeoLife GPS değişken uyumluluğu

| Değişken          | Kaynak alan | Uyumluluk     | Not                                         |
|-------------------|-------------|---------------|---------------------------------------------|
| Coğrafi           |             |               |                                             |
| distance          | lat, lon    | Türetilebilir | Merkeze göre haversine benzeri mesafe (km). |
| terrain_type      | —           | Yok           | Kod sabiti 5.0.                             |
| terrain_harshness | —           | Yok           | Kod sabiti 5.0.                             |
| weather           | —           | Yok           | Kod sabiti 5.0.                             |

|                                    |             |                         |                                              |
|------------------------------------|-------------|-------------------------|----------------------------------------------|
| user_density                       | lat         | Türetilir               | Enlem mutlak değeri.                         |
| Görev                              |             |                         |                                              |
| task_type                          | —           | Yok                     | Adaptör<br>`COMPUTATION`.                    |
| task_size                          | hız (speed) | Türetilir               | speed×50 — mobilite<br>türevli görev boyutu. |
| task_complexity                    | hız         | Türetilir               | Segment hızı.                                |
| recurrence                         | —           | Yok                     | Kod sabiti 3.0.                              |
| delay_tolerance                    | —           | Yok                     | Kod sabiti 50.0.                             |
| Yerel                              |             |                         |                                              |
| cpu_power                          | —           | Yok                     | Kod sabiti 70.                               |
| memory                             | —           | Yok                     | Kod sabiti 70.                               |
| hdd_space                          | —           | Yok                     | Kod sabiti 70.                               |
| network_adapter_speed              | —           | Yok                     | Kod sabiti 70.                               |
| bus_speed                          | —           | Yok                     | Kod sabiti 70.                               |
| Sunucu                             |             |                         |                                              |
| server_type                        | —           | Türetilir               | `server_edge_indicator`<br>0.5.              |
| cpu_power_availability             | —           | Yok                     | Kod sabiti 70.                               |
| memory_availability                | —           | Yok                     | Kod sabiti 70.                               |
| hdd_space_availability             | —           | Yok                     | Kod sabiti 70.                               |
| bus_speed_availability             | —           | Yok                     | Kod sabiti 70.                               |
| network_adapter_speed_availability | —           | Yok                     | Kod sabiti 70.                               |
| network_queue                      | —           | Yok                     | Kod sabiti 10.                               |
| mobility                           | hız         | Doğrudan /<br>Türetilir | GPS segment hızı —<br>**güçlü alan**.        |
| speed                              | hız         | Doğrudan /<br>Türetilir | Sunucu hız vektörü.                          |
| Ağ                                 |             |                         |                                              |
| connection_type                    | —           | Yok                     | Kod sabiti 2.0.                              |
| connection_speed                   | hız         | Türetilir               | speed×200.                                   |
| connection_reliability             | —           | Yok                     | Kod sabiti 75.                               |
| node_count                         | —           | Yok                     | Kod sabiti 4.0.                              |
| Dış Ortam                          |             |                         |                                              |
| user_speed                         | hız         | Doğrudan /<br>Türetilir | Kullanıcı hızı.                              |
| user_direction                     | lat, lon    | Türetilir               | (lat×1000+lon×1000)<br>mod 360.              |
| crowd_frequency                    | hız         | Türetilir               | speed×5.                                     |
| crowd_usage                        | —           | Yok                     | Kod sabiti 30.                               |

Özet: Doğrudan/Türetilir: ~10 (mobilite/coğrafi ağırlıklı), Yok: ~22. \*\*Güçlü boyut:\*\* mesafe, hız, yön, mobilite.

INFOCOM'23 5G Ölçümü (5G\_MONARCH)

Kaynak: Liu ve Peng (2023). \*\*Ham alanlar:\*\* grid lat/lon, toplam ölçüm, Sub-6 GHz, 5G, LTE kanal güçleri (CSV sütun 0–10).

INFOCOM'23 5G Ölçümü değişken uyumluluğu

| Değişken                           | Kaynak alan          | Uyumluluk               | Not                                     |
|------------------------------------|----------------------|-------------------------|-----------------------------------------|
| Coğrafi                            |                      |                         |                                         |
| distance                           | lat, lon             | Türetilbilir            | Grid merkezine göre mesafe (km).        |
| terrain_type                       | -                    | Türetilbilir            | Enlem.                                  |
| terrain harshness                  | -                    | Türetilbilir            | Boylam.                                 |
| weather                            | —                    | Yok                     | Kod sabiti 5.0.                         |
| user_density                       | total (ölçüm sayısı) | Türetilbilir            | Grid yoğunluğu.                         |
| Görev                              |                      |                         |                                         |
| task_type                          | —                    | Yok                     | Adaptör `COMPUTATION`.                  |
| task_size                          | total                | Türetilbilir            | total×10.                               |
| task_complexity                    | 5G + Sub-6           | Türetilbilir            | Kanal karmaşıklığı.                     |
| recurrence                         | —                    | Yok                     | Kod sabiti 3.0.                         |
| delay_tolerance                    | 5G gücü              | Türetilbilir            | (1-g5)×100 — düşük 5G → düşük tolerans. |
| Yerel                              |                      |                         |                                         |
| cpu_power                          | —                    | Yok                     | Kod sabiti 75.                          |
| memory                             | —                    | Yok                     | Kod sabiti 75.                          |
| hdd_space                          | —                    | Yok                     | Kod sabiti 70.                          |
| network_adapter_speed              | 5G gücü              | Türetilbilir            | g5×100.                                 |
| bus_speed                          | —                    | Yok                     | Kod sabiti 70.                          |
| Sunucu                             |                      |                         |                                         |
| server_type                        | —                    | Türetilbilir            | `server_edge_indicator` 0.7.            |
| cpu_power_availability             | —                    | Yok                     | Kod sabiti 70.                          |
| memory_availability                | —                    | Yok                     | Kod sabiti 70.                          |
| hdd_space_availability             | —                    | Yok                     | Kod sabiti 70.                          |
| bus_speed_availability             | —                    | Yok                     | Kod sabiti 70.                          |
| network_adapter_speed_availability | 5G gücü              | Türetilbilir            | g5×100.                                 |
| network_queue                      | LTE                  | Türetilbilir            | lte×30.                                 |
| mobility                           | —                    | Yok                     | Kod sabiti 2.0.                         |
| speed                              | —                    | Yok                     | Kod sabiti 2.0.                         |
| Ağ                                 |                      |                         |                                         |
| connection_type                    | 5G gücü              | Türetilbilir            | g5×4+1 — **güçlü alan**.                |
| connection_speed                   | 5G, LTE              | Doğrudan / Türetilbilir | g5×8000 + lte×500.                      |
| connection_reliability             | 5G, Sub-6            | Türetilbilir            | (g5+sub6)×50+40.                        |
| node_count                         | total                | Türetilbilir            | total/100.                              |
| Dış Ortam                          |                      |                         |                                         |
| user_speed                         | —                    | Yok                     | Kod sabiti 3.0.                         |
| user_direction                     | lat                  | Türetilbilir            | lat mod 360.                            |
| crowd_frequency                    | total                | Türetilbilir            | Ölçüm yoğunluğu.                        |
| crowd_usage                        | LTE                  | Türetilbilir            | lte×100.                                |

Özet: Doğrudan/Türetilebilir: ~18 (ağ ağırlıklı), Yok: ~14. **\*\*Güçlü boyut:\*\*** bağlantı hızı, güvenilirlik, kanal tipi.

MLPerf / CNN-DailyMail (MLPERF\_EDGE)

Kaynak: Hermann vd. (2015); Reddi vd. (2020). **\*\*Ham alanlar:\*\*** CNN/DailyMail `article` metin uzunluğu (çıkartım/NLP iş yükü vektörü).

MLPerf / CNN-DailyMail değişken uyumluluğu

| Değişken                           | Kaynak alan                 | Uyumluluk                | Not                                     |
|------------------------------------|-----------------------------|--------------------------|-----------------------------------------|
| Coğrafi                            |                             |                          |                                         |
| distance                           | article uzunluğu            | Türetilebilir            | Metin boyutu → mesafe vektörü.          |
| terrain_type                       | -                           | Türetilebilir            |                                         |
| terrain harshness                  | —                           | Yok                      | Kod sabiti 5.0.                         |
| weather                            | —                           | Yok                      | Kod sabiti 5.0.                         |
| user_density                       | article uzunluğu mod 200    | Türetilebilir            |                                         |
| Görev                              |                             |                          |                                         |
| task_type                          | —                           | Yok                      | Adaptör `COMPUTATION`.                  |
| task_size                          | article uzunluğu            | Doğrudan / Türetilebilir | <b>**Güçlü alan**</b> — NLP yük boyutu. |
| task_complexity                    | $\log_{1p}(\text{uzunluk})$ | Türetilebilir            | Metin karmaşıklığı.                     |
| recurrence                         | —                           | Yok                      | Kod sabiti 3.0.                         |
| delay_tolerance                    | article uzunluğu            | Türetilebilir            | uzunluk/100.                            |
| Yerel                              |                             |                          |                                         |
| cpu_power                          | —                           | Yok                      | Kod sabiti 75.                          |
| memory                             | —                           | Yok                      | Kod sabiti 75.                          |
| hdd_space                          | —                           | Yok                      | Kod sabiti 70.                          |
| network_adapter_speed              | —                           | Yok                      | Kod sabiti 80.                          |
| bus_speed                          | —                           | Yok                      | Kod sabiti 75.                          |
| Sunucu                             |                             |                          |                                         |
| server_type                        | —                           | Türetilebilir            | `server_edge_indicator` 0.65.           |
| cpu_power_availability             | —                           | Yok                      | Kod sabiti 70.                          |
| memory_availability                | —                           | Yok                      | Kod sabiti 70.                          |
| hdd_space_availability             | —                           | Yok                      | Kod sabiti 70.                          |
| bus_speed_availability             | —                           | Yok                      | Kod sabiti 70.                          |
| network_adapter_speed_availability | —                           | Yok                      | Kod sabiti 75.                          |
| network_queue                      | —                           | Yok                      | Kod sabiti 10.                          |
| mobility                           | —                           | Yok                      | Kod sabiti 1.0.                         |
| speed                              | —                           | Yok                      | Kod sabiti 1.0.                         |

|                        |                      |               |                 |
|------------------------|----------------------|---------------|-----------------|
| Ağ                     |                      |               |                 |
| connection_type        | —                    | Yok           | Kod sabiti 2.5. |
| connection_speed       | article uzunluğu     | Türetilebilir | uzunluk×2.      |
| connection_reliability | article uzunluğu mod | Türetilebilir | mod 30 + 70.    |
| node_count             | —                    | Yok           | Kod sabiti 4.0. |
| Dış Ortam              |                      |               |                 |
| user_speed             | —                    | Yok           | Kod sabiti 2.0. |
| user_direction         | —                    | Yok           | Kod sabiti 180. |
| crowd_frequency        | —                    | Yok           | Kod sabiti 30.  |
| crowd_usage            | —                    | Yok           | Kod sabiti 30.  |

Özet: Doğrudan/Türetilebilir: ~10 (görev/ağ ağırlıklı), Yok: ~22. **\*\*Güçlü boyut:\*\*** task\_size, task\_complexity, gecikme hassas çıkarım yükü.

Aşağıdaki özet çizelge, beş veri setinin MEC simülasyonu bağlamında karşılaştırmalı profilini sunmaktadır.

#### Veri setleri genel karşılaştırması

| Özellik          | Alibaba v2018      | Azure VM       | GeoLife GPS  | INFOCOM'23 5G | MLPerf/CNN-DM    |
|------------------|--------------------|----------------|--------------|---------------|------------------|
| Kod etiketi      | ALIBABA_V2         | AZURE_PUBLIC   | CRAWDA       | 5G_MONARCH    | MLPERF_EDGE      |
| Odağı            | Küme iş yükü       | VM yerleştirme | GPS mobilite | Saha 5G grid  | NLP çıkarım yükü |
| Coğrafi (5)      | Zayıf (türetilmiş) | Zayıf          | Güçlü        | Orta          | Zayıf            |
| Görev (5)        | Güçlü              | Orta           | Zayıf        | Zayıf         | Güçlü            |
| Yerel (5)        | Orta               | Orta           | Sabit        | Sabit         | Sabit            |
| Sunucu (9)       | Güçlü              | Güçlü          | Zayıf        | Zayıf         | Zayıf            |
| Ağ (4)           | Zayıf              | Orta           | Orta         | Güçlü         | Orta             |
| Dış ortam (4)    | Sabit              | Sabit          | Güçlü        | Orta          | Sabit            |
| 32 değişken uyum | Kısmi (~20)        | Kısmi (~16)    | Kısmi (~10)  | Kısmi (~18)   | Kısmi (~10)      |
| Tezde kullanım   | Evet               | Evet           | Evet         | Evet          | Evet             |